

2014

Instituto Politécnico de Coimbra

INSTITUTO SUPERIOR DE ENGENHARIA DE COIMBRA

IBackOffice

MESTRADO EM INFORMÁTICA E SISTEMAS

AUTOR | Carlos Guilherme Fontes Tomás

ORIENTADORES | Prof.^a Doutora Fernanda Barbosa
Eng.^o Ricardo Teixeira

Coimbra, dezembro 2014



Instituto Superior de
Engenharia de Coimbra

Instituto Politécnico de Coimbra
Instituto Superior de Engenharia de Coimbra
Departamento de Engenharia Informática e Sistemas

Mestrado em Informática e Sistemas

Desenvolvimento de Software

Relatório final de estágio

IBackOffice

Carlos Guilherme Fontes Tomás [a21180811@alunos.isec.pt]

Estágio sob orientação:

Fernanda Correia Barbosa (IPC)

Ricardo Teixeira (Kamae)

Coimbra, Dezembro, 2014

Agradecimentos

Quero agradecer aqueles que acreditaram em mim e que por seu sacrifício e vontade eu pude seguir este caminho e estar aqui, obrigado família.

Quero agradecer aos colegas e amigos pelo apoio dado.

Quero agradecer aqueles que me transmitiram o conhecimento e a sabedoria para o poder procurar obrigado professores.

Quero agradecer à equipa da Kamae por me terem acolhido e apoiado na realização deste projecto.

Quero agradecer à Professora Fernanda Correia Barbosa por me ter orientado durante a realização deste projecto.

Resumo

Este estágio foi realizado no âmbito da unidade curricular “Estágio ou Projecto Industrial”, pertencente ao Mestrado em Engenharia Informática e Sistemas - Desenvolvimento de Software, do Instituto Superior de Engenharia de Coimbra do Instituto Politécnico de Coimbra.

O estágio foi realizado na empresa Kamae, que tem vindo a desenvolver jogos para o mercado do *iOS* e mais recentemente tem vindo a desenvolver algumas aplicações mais voltadas para a advocacia tanto para *iOS* como para *android*.

O estágio com o título “IBackOffice”, consiste na criação de um *site* para o controlo de aplicações *iPhone* desenvolvidas na empresa, tendo como principais funcionalidades a possibilidade de consultar dados das aplicações da empresa e o envio de notificações para o utilizador.

Durante o decorrer do estágio foram realizadas diversas actividades. Numa primeira etapa foi feita a análise de requisitos e criação de *mockups* para a aplicação desenvolvida e numa segunda etapa foi feito o desenvolvimento da aplicação que envolveu a criação de provas de conceito, desenvolvimento, integração de funcionalidades e por fim correcção de *bugs* reportados (os *bugs* resultaram da fase de testes feita por outro elemento da Kamae).

Neste relatório é feita uma descrição da empresa Kamae e de como foi utilizada a sua metodologia de desenvolvimento de *software* o *scrum*, é feita uma descrição da aplicação “IBackOffice” e são analisadas as tecnologias utilizadas, sendo as principais: *razor*, *entity framework*, *LINQ*, *infragistics* e *jquery*.

Palavras-chave: *IPhone*, aplicação, *razor*, *scrum*, *nav screen*, *more screen*, *push-notifications*.

Abstract

This internship was held under the course of " *Estágio ou Projecto Industrial* ", belonging to the master degree in Informatics Engineering and Systems - Software Development from the *Instituto Superior de Engenharia de Coimbra* of the *Instituto Politécnico de Coimbra*.

The internship was conducted in the *Kamae* company, which has been developing some games for the iOS market, and more recently has been developing applications focused on advocacy subject, both for iOS and android.

The internship with the title "IBackOffice" is the development of a website for the control of the iPhone applications made in the company *Kamae*, and has as main features the ability to consult applications data and send notifications to the user.

During the internship several activities were carried out. First stage included the requirements analysis and the creation of mockups and a second stage that included the development involving creation of proofs of concept, the development and the integration of features and finally the correction of reported bugs (these bugs resulted from a testing phase performed by another element of *Kamae*).

In this report is done a description of the *Kamae* company and of the use of the scrum as the software development methodology, description of the "IBackOffice" application and the analysis of the main technologies used like, razor, entity framework, LINQ, infragistics and jQuery.

Key-words: iPhone, application, razor, scrum, nag screen, more screen, push-notifications.

Índice

Agradecimentos	i
Resumo	ii
Abstract.....	iii
Índice de Figuras	vi
Índice de Tabelas	vii
Significados e Abreviaturas.....	viii
1. INTRODUÇÃO.....	1
1.1. Contextualização.....	1
1.2. Entidades envolvidas	2
1.2.1. ISEC	2
1.2.2. Kamae RT.....	2
1.3. Coordenação	3
1.4. Objectivos do estágio	3
1.5. Estrutura do relatório	4
2. Implementação.....	5
2.1. Estado do projecto	5
2.2. Elaboração de <i>mockups</i> e análise de requisitos.....	5
2.3. Arquitectura do IBO	6
2.4. Globalização	9
2.4.1. Resources.....	10
2.4.2. Conversão de moedas	10
2.5. Ferramentas utilizadas	11
2.6. Tecnologias	12
2.6.1. Razor.....	12
2.6.2. MVC.....	12
2.6.3. <i>Javascript</i>	13
2.6.4. LINQ.....	13
2.6.5. Lambda	13
2.6.6. Entity framework.....	14
2.6.7. Infragistics	14
2.6.8. Web services.....	15
2.6.9. JSON.....	15
2.6.10. XML.....	15
2.6.11. CSS	16
2.6.12. Ajax.....	16
2.6.13. C#.....	16

2.6.14.	Layout	17
2.6.15.	Partialview	17
2.6.16.	Membership	17
2.6.17.	Forms authentication.....	17
2.6.18.	Metro Js.....	17
2.6.19.	Multiple select.....	17
2.6.20.	Itunes Search API	17
2.6.21.	Chartboost Api.....	17
2.6.22.	APNs.....	18
2.6.23.	PushSharp	18
2.6.24.	Regex	18
2.6.25.	DataAnnotations	18
2.7.	Páginas do IBO	20
2.7.1.	Página de Login	20
2.7.2.	Página de registo.....	21
2.7.3.	Página principal	21
2.7.4.	Página das apps.....	22
2.7.5.	Página de <i>More screens</i>	26
2.7.6.	Página de <i>nag screens</i>	28
2.7.7.	Página <i>Push notifications</i>	29
2.7.8.	Página de <i>sales and trends</i>	30
2.7.9.	Página de <i>outsource</i>	32
2.7.10.	Página de opções.....	33
2.8.	Integração com as aplicações.....	34
3.	Análise de dados	37
4.	Desenvolvimento na Kamae	39
4.1.	Aplicação de metodologia <i>Scrum</i> no estágio	39
4.2.	Ferramentas.....	40
4.3.	Partilha de conhecimentos	41
5.	Conclusões.....	42
5.1.	Análise do desenvolvimento da aplicação IBO	42
5.2.	Metodologia	42
5.3.	Trabalho futuro	42
	Bibliografia	44

Índice de Figuras

Figura 1 - Esquema global da aplicação	7
Figura 2 - Esquema MVC e componentes da vista	8
Figura 3 - Esquema <i>push notifications</i>	9
Figura 4 - Esquema MVC.....	12
Figura 5 – Exemplo de código com LINQ.	13
Figura 6 – Exemplo de código Lambda.....	14
Figura 7 – Exemplo (Select) no <i>Entity Framework</i> [24]	14
Figura 8 – Exemplo de conteúdo de ficheiro JSON	15
Figura 9 – Exemplo de classe que recebe os dados do ficheiro JSON da figura 8.....	15
Figura 10 – Exemplo de código para obter o JSON para as classes.....	15
Figura 11 - Exemplo de ficheiro em XML	16
Figura 12 – Exemplo de código em AJAX	16
Figura 13 – Exemplo de uma expressão regular [26] para números entre 200 e 249	18
Figura 14 – Exemplo de <i>DataAnotations</i> do lado do servidor	19
Figura 15 – Exemplo de <i>DataAnotations</i> do lado do cliente.....	19
Figura 16 - Página de <i>login</i>	20
Figura 17 - Página de registo.....	21
Figura 18 - Página principal (<i>Home page</i>)	22
Figura 19 - Página das aplicações do site IBO.	22
Figura 20 – Exemplo de <i>dropdown</i> de <i>check boxes</i>	23
Figura 21 - Formulário para inserir uma aplicação no site IBO	23
Figura 22 - Parte da grelha das aplicações.	24
Figura 23 - Ver aplicação no <i>site IBO</i> (parte 1)	24
Figura 24 - Ver aplicação no <i>site IBO</i> (parte 2)	25
Figura 25 - Formulário para fazer <i>upload</i> de um ficheiro para <i>push notifications</i> no <i>site IBO</i>	25
Figura 26 - Formulário para inserir detalhes de uma aplicação no <i>site IBO</i>	26
Figura 27 - Pagina <i>More screens</i> do <i>site IBO</i>	27
Figura 28 - Formulário para inserir <i>more screens</i> no <i>site IBO</i>	27
Figura 29 - Página de <i>nag screens</i> do <i>site IBO</i>	28
Figura 30 - Formulário para inserir <i>nag screens</i> no <i>site IBO</i>	29
Figura 31 - Página de <i>push notifications</i> do <i>site IBO</i>	29
Figura 32 - Formulário para enviar <i>push notifications</i>	30
Figura 33- Página <i>Sales and Trends</i> do <i>site IBO</i> (parte 1)	31
Figura 34 - Página <i>Sales and Trends</i> do <i>site IBO</i> (parte 2)	31
Figura 35 - Página <i>sales and trends</i> do <i>site IBO</i> (parte 3).....	32
Figura 36 - Página de <i>outsource</i> do <i>site IBO</i>	32
Figura 37 - Formulário para inserir um <i>outsource</i> no <i>site IBO</i>	33
Figura 38 - Página de opções do <i>site IBO</i>	33
Figura 39 - Formulário para introduzir credenciais do <i>Apple connect</i> do <i>site IBO</i>	34
Figura 40 - Formulário para inserir credenciais no <i>chartboost</i> do <i>site IBO</i>	34
Figura 41 - <i>Nag screen iPhone</i>	35
Figura 42 – Imagem de <i>push notification</i>	35
Figura 43 - Esquema de metodologia <i>scrum</i> no <i>estágio</i>	40

Índice de Tabelas

Tabela 1 – Ferramentas utilizadas	11
---	----

Significados e Abreviaturas

ADM - *Amazon Device Messaging*.

AJAX - *Asynchronous Javascript and XML*.

APNs - *Apple Push Notification service*.

App – Aplicação.

ASP - *Active Server Pages*.

Bundle Id - O *bundle identifier* é um ID para a aplicação, utilizado pelo sistema para que este possa guardar informações e referenciar-se à aplicação de forma única.

C2DM - *Android Cloud to Device Messaging*.

CSS - *Cascading Style Sheets*.

DLL – *Dynamic link library*.

HTML - *HyperText Markup Language*.

IBO – IBackOffice.

Id de vendedor - *Vendor id* poderá ser obtido no *itunes connect*.

Inapps - São objectos que podem ser comprados dentro da aplicação.

JSON - *JavaScript Object Notation*.

Kamae – Kamae é uma postura das artes marciais que além de denotar como o lutador se posiciona, aponta como o lutador se comporta e o seu estado de espírito este deve estar preparado para agir e reagir conforme cada circunstância: Kamae não está no corpo mas na mente.

Mockup - O *mockup* é uma representação bem elaborada de um *design* que servira de protótipo do *interface*. Nele podendo ser representados botões, cores e fontes.

More screens - Ecrã na aplicação onde são mostrados outros produtos da empresa.

MVC - *Model-view-controller*.

Nag screens - Um *nag screen* é um ecrã que é mostrado ao utilizador para encorajar o utilizador de uma aplicação a comprar algum produto.

PAP - *Push Access Protocol*.

Push notifications - Uma *push notifications* é um tipo de notificação enviada de um servidor que permite notificar o utilizador de uma aplicação de algum evento, novidade e outros assuntos.

SKU - *Stock Keeping Unit*.

TFS - *Team Foundation Server*.

URL - *Uniform Resource Locator*.

XML - *Extensible Markup Language*.

1. INTRODUÇÃO

O ramo de Desenvolvimento de Software do Mestrado em Engenharia Informática e Sistemas do Instituto Superior de Engenharia de Coimbra visa a preparação de profissionais competentes no exercício profissional, no domínio de desenvolvimento de *software*, em todos as suas etapas, incluindo os aspectos importantes de gestão de projectos, equipas e garantia de qualidade.

O estágio final, sendo desenvolvido no ambiente das empresas de acolhimento e acompanhado pelos docentes do mestrado, tem como principal objectivo criar um espaço de transição entre a vida estudantil e a vida profissional, onde o aluno possa adquirir experiência de trabalho, melhorar os seus conhecimentos e habilidades e tenha a possibilidade de aplicar os conhecimentos adquiridos durante a realização do curso.

Para que o aluno realize um estágio, foi proposto um estágio na empresa Kamae onde o aluno desenvolveu o projecto “IBackOffice”, uma plataforma web que de forma resumida se destina a consultar informações das aplicações desenvolvidas pela empresa com a possibilidade de envio de notificações para os utilizadores.

1.1. Contextualização

Actualmente há uma grande utilização de *smartphones* um pequeno computador que pode viajar nos nossos bolsos e utilizado em qualquer hora, demonstrando-se então um mercado interessante a ser explorado pelas empresas de *software*. Outro mercado importante a ser levado em conta é a utilização de jogos que ocupam grande fatia dos *downloads* nos mercados de aplicações e também do tempo gasto pelos utilizadores de *smartphones*. A empresa Kamae desenvolve aplicações neste mercado das aplicações móveis quer na área de advocacia, quer no mercado dos jogos. A empresa já conta com cerca de 14 aplicações sendo a sua maioria jogos.

Algumas das formas de obter lucro com aplicações será através de publicidade, da venda de *inapps* e da venda da própria aplicação. Então, agora, levanta-se o problema de como levar as aplicações aos utilizadores, como fazer com que estas sejam utilizadas e como fazer com que o utilizador não se esqueça delas aquando da sua instalação e como publicitar as aplicações. A resposta encontrada são as *push notifications*, *nag screens* e *more screens* que irão dar uma melhor interacção com o utilizador final e ao mesmo tempo tornar as aplicações mais rentáveis.

Na internet, existe uma enorme variedade de serviços que podem ser utilizados para fazer publicidade às aplicações e também para obter informações destas, o problema é que essa informação está dispersa por várias plataformas tornando-se, então, importante centralizar essa informação e ao mesmo tempo mantê-la para que possa ser manipulada e para que se possa obter mais informação.

É neste contexto que surge o IBackOffice, uma vez que há uma enorme necessidade de possuir tanta informação quanto possível na “palma da mão”. O IBackOffice surge para gerir as aplicações em determinadas áreas, proporcionar mais interacção com o utilizador final,

obter uma informação rápida e actualizada sobre os aplicativos e para reagir rapidamente às tendências de mercado.

1.2. Entidades envolvidas

1.2.1. ISEC

“O Instituto Superior de Engenharia de Coimbra (ISEC) foi fundado em 1965, resultou da conversão do antigo Instituto Industrial e Comercial de Coimbra. Mais tarde, em 1988, o ISEC foi integrado no Ensino Superior Politécnico. O lema do ISEC é inovação, sucesso, empreendedorismo e criatividade. Actualmente são leccionados cursos de especialização tecnológica, licenciaturas, mestrados e pós-graduações em diferentes áreas da engenharia”[29].

Missão- “O ISEC tem como missão a criação, transmissão e difusão de cultura, ciência e tecnologia, cabendo-lhe ministrar uma formação de nível superior para o exercício de actividades profissionais no domínio da Engenharia e promover o desenvolvimento da região em que se insere”.

Visão- “O ISEC tem como visão institucional ser uma referência de excelência no ensino, reconhecido nacional e internacionalmente por serviços de qualidade e relevância social, com práticas flexíveis, criativas e inovadoras. Pretende ainda ser um parceiro privilegiado das organizações empresariais e das famílias da região onde se insere pela orientação eminentemente prática, fundada num rigoroso conhecimento teórico, que imprime a todas as suas actividades”.

Valores- “Os valores fundamentais pelos quais se rege o ISEC são a cidadania, a qualidade, a busca constante da valorização, motivação e actualização pedagógica, científica e tecnológica dos seus recursos, o bom relacionamento e a disponibilidade para com os estudantes e as organizações suas parceiras e a preocupação com o desenvolvimento social e económico da região onde está inserido”.

1.2.2. Kamae RT

A Kamae RT encontra-se sediada em Fala (Rua da Fontinha, nº8, Fala 3045-398 Coimbra), foi criada em 1997 e é uma empresa que desenvolve soluções informáticas. Conta já com 17 anos e uma vasta experiência em desenvolvimento de sistemas jurídicos e de gestão. Começou pela venda de *hardware*, desenvolvimento de sites web e consultadoria informática para empresas nacionais. Hoje é uma empresa que conta com mais de 20 produtos comercializados no mercado e vários prémios ganhos ao longo da sua existência. A empresa procura incorporar o seu espírito nos seus produtos, colaboradores e ambiente onde actua.

Missão- “Aumentar a produtividade do negócio dos nossos clientes através do desenvolvimento de Sistemas de Informação parciais ou integrais”[30].

Valores- A empresa apresenta um espírito de guarda e honra intimamente ligado às artes marciais e guerreiros samurais que valorizam a perfeição, metodologia, força e segurança.

1.3. Coordenação

Durante o decorrer do estágio na empresa foram realizadas reuniões todos os dias (*daily scrum meetings*) com a restante equipa de desenvolvimento e um membro do suporte. Foram ainda realizadas reuniões de final de *sprint* e reuniões para apresentação do projecto.

Nas reuniões diárias era apresentado o trabalho realizado desde da última reunião, objectivos do próximo dia e desafios encontrados.

Nas apresentações de *sprints* eram apresentados os objectivos cumpridos e definidos os objectivos do próximo *sprint*. Esta reunião era feita com toda a equipe de desenvolvimento.

A apresentação do projecto era feita com o estagiário e um membro da empresa onde é apresentado o projecto em si. Desta reunião resulta o feedback relativo às novas funcionalidades implementadas na aplicação (se realmente as coisas funcionam assim como está feito e se alguma funcionalidade tem de ser melhorada).

1.4. Objectivos do estágio

A realização do estágio teve como objectivo implementar as seguintes funcionalidades na plataforma IBackOffice.

- Versão 1.0
 - Registo iApps;
 - Histórico de palavras-chave e correspondentes resultados;
 - *Push-notifications*;
 - *More Screen*;
 - *Nag Screens*;
 - Controlo de publicidade: activar e desactivar;
- Versão 2.0
 - *Resources* para vários idiomas;
 - *Tops*;
 - Alertas de *mails*;
- Versão 3.0
 - Análise de Palavras-chave;
 - Tendências de Mercado;
 - Análise de comportamentos;
- Versão 4.0:
 - Dados Financeiros;
 - *Google Play*;

1.5. Estrutura do relatório

O relatório de estágio encontra-se dividido em seis capítulos principais.

O primeiro capítulo tem como objectivo fazer uma introdução ao estágio, indicar os seus objectivos, o contexto no qual é realizado apresentar a empresa de acolhimento e a instituição de ensino.

O segundo capítulo tem como objectivo fazer uma apresentação da aplicação, em termos do seu desenvolvimento, arquitectura, tecnologias utilizadas, aparência e funcionalidades.

O terceiro capítulo tem como objectivo apresentar as bases de dados utilizadas no projecto o seu esquema e como foram criadas.

O quarto capítulo tem como objectivo apresentar a metodologia utilizada no desenvolvimento de *software* na kamae.

O quinto capítulo tem como objectivo apresentar as conclusões.

O sexto capítulo são as referências bibliográficas.

2. Implementação

A aplicação *IBO* foi desenvolvida para a web sendo um *web site* que deverá facilitar o controlo das aplicações da empresa. Devendo funcionar nos *browsers internet Explorer, Google chrome e firefox*.

Antes de iniciar o desenvolvimento foi feita uma análise de requisitos e através destes requisitos foram criados *mockups* (ver cd de anexos pasta de *mockups*).

Para o controlo dos requisitos, foi utilizado o *visual studio online*[31] e mais tarde para o controlo de *bugs* o TFS da Microsoft[32].

A aplicação *IBackOffice* foi desenvolvida no *visual studio 2012* em *asp.net* utilizando o *view engine razor*.

2.1. Estado do projecto

Quando o estágio foi iniciado o projecto já se encontrava em desenvolvimento. As páginas de login, *home*, *nag screens* e *push notifications* já se encontravam implementadas, mas no entanto sofreram alterações. Por exemplo, na página de login a autenticação era estática, apenas aceitava um login estático e passou a ser feita dinamicamente, tendo apenas sido aproveitada em termos de *design*.

Da página *home*, que tinha um gráfico com informação dos cliques nos *nag screens*, foi utilizado o *layout* que também se aplica às outras páginas.

Na página de aplicações, que permitia adicionar e actualizar aplicações manualmente, as aplicações encontravam-se dispostas numa lista com todas as aplicações e as suas informações, estando agora estas dispostas em grelha.

Na página para adicionar *nag screens* as aplicações encontravam-se dispostas em listas e continham um quadro com todos os *nag screens*. Era possível adicionar, eliminar e editar os *nag screens*.

A página para enviar *push notifications* foi reaproveitada. Já se encontravam desenvolvidas as formas para enviar *push notications*, mas tiveram de ser refeitas pois a biblioteca que estava a permitir enviar notificações já estava desactualizada.

As páginas de *nag screen* e aplicações foram refeitas para suportar mais informação.

2.2. Elaboração de *mockups* e análise de requisitos

A primeira etapa do projecto consistiu na criação da documentação de forma que o estagiário tivesse um guia para a realização do projecto e mais do que isso entender como funcionava a empresa, entender o projecto em que estava inserido (não só no contexto em que estava inserido mas também as tecnologias a utilizar e plataformas com as quais iria interagir) e por fim para uma melhor integração no meio da empresa. A documentação inicial envolveu a

realização de uma pesquisa que resultou na criação de um documento de requisitos e na elaboração de *mokups*.

Foi realizada uma pesquisa inicial para adquirir conhecimentos em *asp razor* (e outras tecnologias que o estagiário havia tido pouco ou nenhum contacto como *javascript*, *css* entre outras). Esta pesquisa envolveu também alguma prática, com a realização de alguns tutoriais e provas de conceito. Em paralelo, foi realizada uma pesquisa de como seriam as interações da aplicação, ou seja como poderiam ser feitos certos elementos da aplicação, que serviços utilizar, se seria possível fazer, como seria possível fazer (isto implicou encontrar tecnologias e bibliotecas que pudessem facilitar a realização do projecto). Do conjunto destas pesquisas resultou o documento de requisitos e um conjunto de fontes onde poderia ir obter informação.

O documento de requisitos foi feito em *Word*. O documento faz uma explicação inicial do projecto a ser implementado, analisa os utilizadores do sistema, apresenta os casos de uso e os requisitos a serem implementados. Os requisitos foram depois registados no *visual studio online* para que o trabalho fosse sendo registrado e controlado.

Os *mockups* foram feitos utilizando a ferramenta *power point storyboarding* e em papel o que ajudava bastante uma vez que existe mais liberdade na criação destes e com a troca de palavras e ideias era mais fácil chegar a um melhor consenso e a uma ideia que se vai aproximar da ideia final. Os *mockups* feitos no *power point storyboarding* eram impressos e entregues para serem revistos, eram feitas correcções ou notas à mão o que facilitava no esclarecimento do trabalho que tinha de ser feito.

2.3.Arquitectura do IBO

O IBO é um *web site* que foi desenhado neste estágio utilizando as tecnologias *asp.net* e segue o padrão MVC. Para além do site principal tem uma *web api* para interagir com as aplicações nos *iPhones* e duas *dll*, *Common dll* para ajudar no mapeamento de classes noutras classes e *Data dll* que contem os objectos criados pela *entity framework* no processo de gerar classes a partir da base de dados e ainda classes que acedem e manipulam os dados vindos da base de dados. Assim se for preciso fazer uma aplicação *desktop* idêntica ao IBO basta copiar a *dll* para o projecto e todo este trabalho estará feito. Um exemplo mais directo será a *web api* desenvolvida que vai precisar fazer acessos a base de dados e para isso bastou adicionar a *dll* ao projecto e foi possível utilizar as funções já existentes sem ter trabalho extra.

Em termos mais directos os utilizadores irão interagir com o *web site* para controlar as suas aplicações e as aplicações irão interagir com a *web api* para obterem *nag screens*, *more screens* e *tokens*, necessários para o envio de *push notifications* e trocas de informações sobre o dispositivo (ver figura 1).

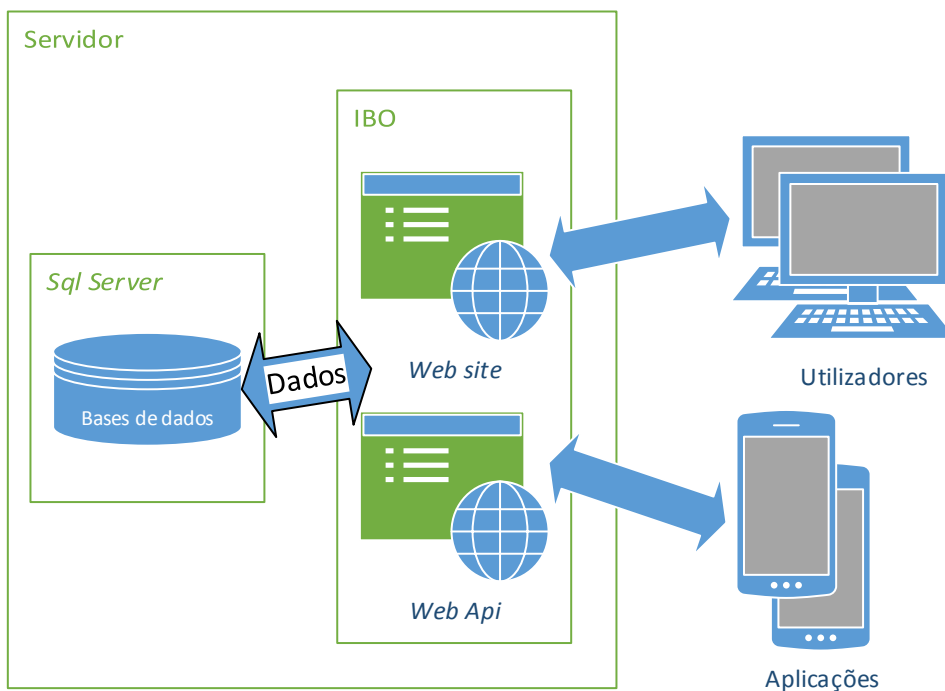


Figura 1 - Esquema global da aplicação

O MVC em ASP funciona da seguinte forma (ver figura 2), um utilizador escreve um URL ou clica num botão do *site* e é feito um pedido (*request*) a um *controller* que por sua vez obtém o modelo (poderá não ser necessário, caso a tarefa seja muito simples) e faz alguma coisa (lógica da aplicação), em seguida, o *controller* envia o modelo para a vista (a vista poderá não receber um modelo, caso seja muito simples e não necessite de um). O trabalho da vista é representar o modelo visualmente (transformar em HTML) e enviar a resposta para o utilizador.

A vista foi construída utilizando várias tecnologias (ver figura 2), uma delas são os “*layouts*”, que permitem definir uma aparência comum para o *site* e será utilizado como herança por várias páginas do *site* e será o equivalente a uma *master page* em *asp*. Foram utilizados *css*, *plugins de jquery* e componentes do *infragistics* para tornar as páginas mais dinâmicas. Por vezes, apenas parte da página teria de ser actualizada e para isso foi utilizada a tecnologia *ajax* que permite actualizar apenas parte da página. Nas situações em que apenas parte da página será actualizada a vista encontra-se dividida em *partial views*, de forma a tornar tudo mais organizado e perceptível. Um determinado controlo, controla a página em si, mas existem outros controlos que por sua vez controlam as *partial views*, se for necessário actualizar parte da página através de *ajax*, serão estes controlos que serão chamados. O *ajax* irá fazer um pedido a um *controller* que por sua vez devolverá a parte da página que foi pedida (normalmente HTML proveniente de uma *partial view/view* que poderá receber outros tipos de dados *XML*, *json* e outros). Por fim, o *ajax* irá actualizar a parte da página pretendida.

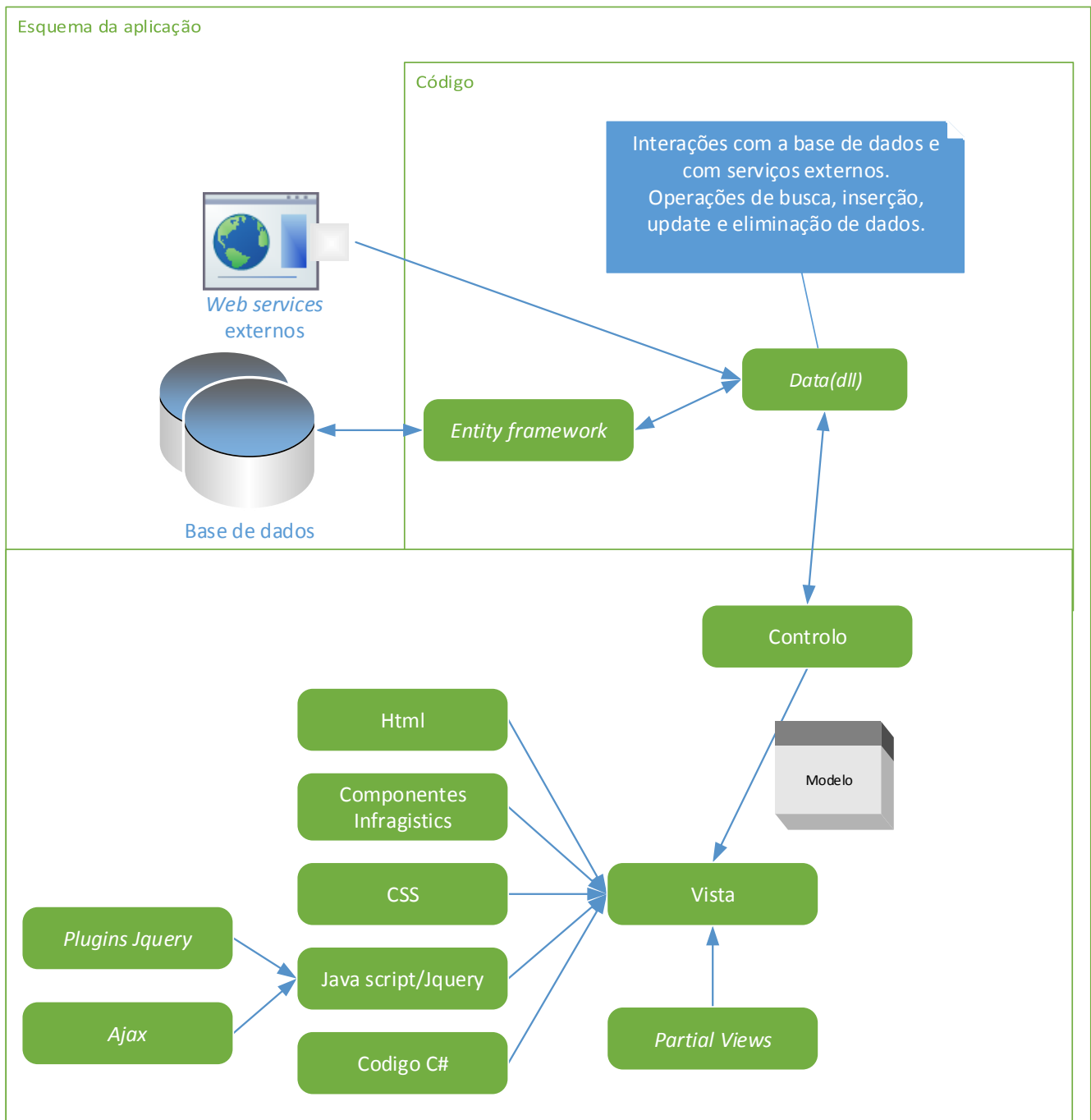


Figura 2 - Esquema MVC e componentes da vista

As *push notifications* são enviadas utilizando a biblioteca *PushSharp* que permite enviar *push notifications* para várias plataformas *iOS*, *android*, *Windows 8*, *windows phone*, entre outras.

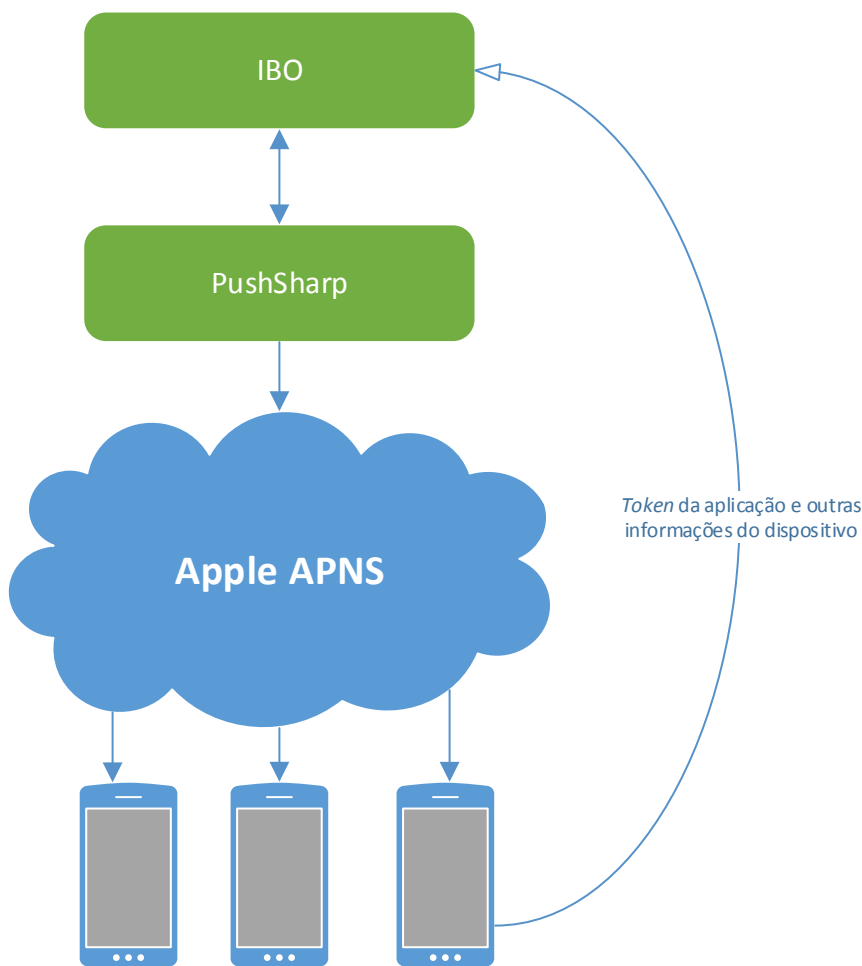


Figura 3 - Esquema *push notifications*

O envio de *push notifications* (ver figura 3) funciona da seguinte forma, o IBO recebe um *token* dos dispositivos com uma determinada aplicação, aquando da primeira utilização desta (o *token* serve para o APNs reconhecer a aplicação e dispositivo para o qual vai enviar a *push notification*), em seguida, quando um utilizador escolher enviar uma *push notification* para uma aplicação, caso esta tenha algum *token* disponível, o IBO envia através do *PushSharp* para o servidor APNs a *push notification* e, por sua vez o servidor APNs irá enviar a notificação para os utilizadores das aplicações (poderá falhar por alguma indisponibilidade do dispositivo ou no caso do *token* ter expirado).

2.4. Globalização

A globalização é o processo de preparação do *site* para que ele seja acessível a um público tão amplo quanto possível. Isto é, em grande parte conseguido através da apresentação de conteúdo na língua nativa do visitante. Para isso as páginas foram feitas utilizando *resources* para as *strings*, visíveis ao utilizador.

2.4.1.Resources

A tecnologia utilizada permite criar ficheiros denominados *Resources*, cuja extensão é “.resx”. Estes ficheiros consistem em entradas XML, podendo ser abertos com um editor de texto normal, de modo a serem manipulados.

O código em C# é capaz de aceder às entradas dos *Resources*, podendo assim ter acesso à informação atrás referida (uma vez que é possível utilizar código C# nas vistas é possível aceder aos *Resources* nestas). Desta forma, sempre que seja necessário mudar esta informação, não é necessário editar o código, apenas o *Resource* correspondente.

A tecnologia usada permite facilmente apresentar as *strings* ao utilizador na sua língua, apenas sendo necessário criar um *Resource* com um nome específico, para cada língua que a aplicação pretende suportar, podendo assim facilmente suportar línguas adicionais.

Posteriormente é necessário, para cada entrada XML atribuir o mesmo nome dos atributos dos ficheiros anteriores, com a respectiva tradução para a língua desejada. A aplicação carregará automaticamente o *Resource* responsável por apresentar os textos na sua língua.

No caso do IBO, quando o utilizador entrar no *site*, a linguagem em que este se encontra será igual à linguagem do browser em que o *site* está aberto ou na linguagem por defeito, o inglês, caso a linguagem em que o browser se encontra não exista nos ficheiros de *Resources*.

Para alterar a linguagem no *site* o utilizador precisa alterar a linguagem por defeito do *browser*.

O site IBO encontra-se traduzido nas línguas inglês e português.

2.4.2.Conversão de moedas

Foi necessário criar uma função que convertesse valores de uma moeda em valores de outra uma vez que muitos dos valores obtidos por exemplo do *itunes connect* se encontravam na moeda em que foram pagas as aplicações. A dificuldade encontra-se no caso de querer mostrar os valores obtidos na totalidade (não se pode somar euros com dólares sem uma conversão).

Para obter os valores das moedas foi necessário aceder a um *web service* que retorna os valores das principais moedas mundiais em relação ao euro [1] este serviço é fornecido pelo banco central europeu e os valores são armazenados num dicionário com código da moeda e valor em relação ao euro. No entanto continuavam a faltar algumas moedas e para isso acedeu-se a um segundo *web service* [2] que permite obter uma maior variedade de moedas. Estes valores serão utilizados como recurso, caso as moedas não existam no primeiro serviço. Os valores são guardados numa Lista com os valores das moedas que são obtidos utilizando a expressão regular $\text{Euro} = ([0-9]^*.[0-9]^*)$ e o código do país.

Foi estabelecido que os valores apresentados estariam representados no IBO em euros. Desta forma a função converte os valores em euros mas esta pode ser facilmente alterada para suportar outras moedas convertendo o valor para euro e depois reconvertendo para o valor da moeda pretendida.

2.5. Ferramentas utilizadas

Na realização do projecto foram utilizadas diversas ferramentas as quais são apresentadas na tabela 1.

Tabela 1 – Ferramentas utilizadas

Ferramenta	Descrição
<i>Microsoft Visual Studio 2012</i>	Ferramenta utilizada para o desenvolvimento do projecto.
<i>Sql server 2014</i> [http://www.microsoft.com/pt-br/server-cloud/products/sql-server/]	Servidor de base de dados.
<i>SQL Server 2014 Configuration Manager</i>	Ferramenta de construção, edição e testes da base de dados. Foi utilizada para construir o esquema gráfico do modelo de dados da base de dados.
<i>SQL Server 2014 Management Studio</i>	Permite ligar e desligar instâncias de bases de dados.
<i>aspnet_regsql.exe</i>	Aplicação utilizada para adicionar o esquema de <i>SqlMembershipProvider</i> a base de dados.
<i>Visual Studio Online (Microsoft)</i> [https://kamae.visualstudio.com/]	Ferramenta utilizada para especificação das funcionalidades a serem desenvolvidas na aplicação. Ferramenta utilizada como repositório do iBO.
Shogun (TFS Microsoft)	Ferramenta utilizada para registo de bugs e controlo do estado dos mesmos.
Portal Kamae [http://sp.kamae.pt/]	Ferramenta para entregar relatórios semanais (dificuldades, o que foi feito semanalmente e os objectivos da próxima semana). Também utilizada para recursos humanos. As faltas devem ser avisadas aqui com antecedência de 7 dias.
<i>Microsoft Visio (2013)</i>	Ferramenta de desenho utilizada para fazer diagramas e esquemas para o projecto.
<i>Microsoft Word 2010</i>	Ferramenta utilizada para fazer a documentação do projecto.
<i>Microsoft PowerPoint 2010</i>	Ferramenta utilizada para fazer as apresentações do projecto. Ferramenta para fazer os <i>mockups</i> .

<i>Syncfusion Metro Studio 2</i>	Criação de ícones utilizados no <i>web site</i> .
http://json2csharp.com/	Esta página <i>web</i> permite inserir dados no formato <i>json</i> e cria uma classe que depois poderá ser utilizada para receber os dados do ficheiro.

2.6. Tecnologias

Para a realização deste projecto foram utilizadas diversas tecnologias, sendo aqui referidas as principais.

2.6.1. Razor

O *Razor* é uma *view engine* que permite incluir a lógica de aplicação (*Visual Basic* e *C #*) na camada de visualização das páginas *web*.

A inclusão de notações *razor*[10] permite criar páginas mais dinâmicas, existindo dois tipos de conteúdo nestas páginas. Uma parte (*cshtml*) que é executada do lado do cliente HTML, CSS e *javascript* e uma parte que é executada do lado do servidor, como por exemplo acesso à base de dados. O *razor* permite que se adicione código do servidor ao conteúdo do cliente, sendo possível numa única página, utilizar HTML em conjunto com código *razor*. A forma de funcionar é simples, o servidor reconhece a extensão CSHTML e então executa em primeiro lugar os códigos do lado do servidor e em seguida envia a página para o *browser* do cliente.

2.6.2. MVC

O MVC[33] é um padrão utilizado em programação que permite construir aplicações baseadas em três camadas: Modelo, vista e controlo (ver figura 4).

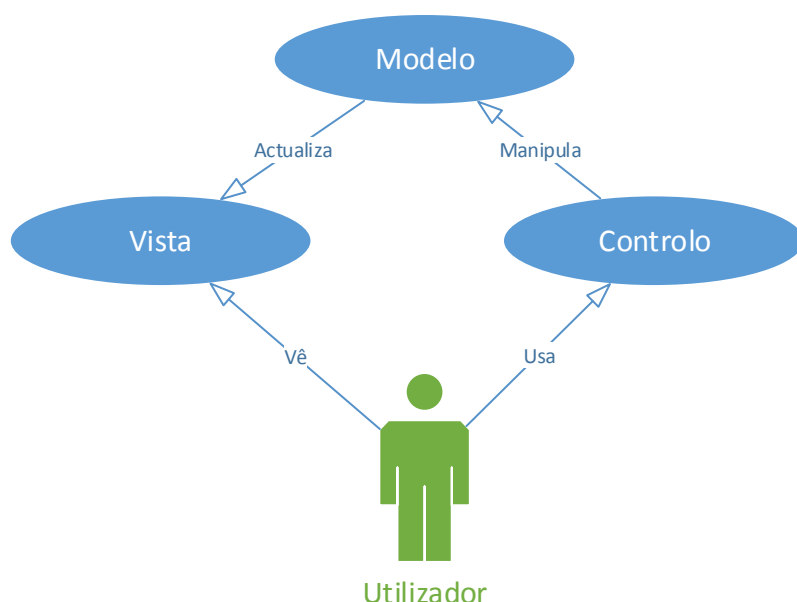


Figura 4 - Esquema MVC

2.6.3. Javascript

Javascript é uma linguagem de programação que corre do lado do cliente sendo interpretada pelos *browsers*. Permite que os *scripts* sejam executados sem passar pelo servidor e permite que parte da página seja alterada (AJAX). O *jquery* [34] é uma biblioteca de *javascript* que facilita a navegação do documento (página HTML), a manipulação de eventos, a utilização do AJAX. Existem imensos *plugins* e elementos gráficos melhorados que podem ser utilizados contribuindo para criar uma aplicação mais rica e dinâmica.

2.6.4. LINQ

Esta tecnologia [35] foi inspirada na linguagem SQL. É usada para consultas em bases de dados relacionais, mas, neste caso, aplicada a colecções de objectos.

Através desta, é possível efectuar consultas em colecções de objectos (*arrays*, listas, dicionários) (ver figura 5). O LINQ permite uma maior eficiência aos programadores, tanto no momento de desenvolvimento como quando se efectua uma análise de código.

O LINQ tem a particularidade de ser executado aquando do acesso à colecção gerada através do seu resultado (execução adiada), o que pode ser vantajoso.

```
static void Main()
{
    int[] numbers = new int[7] { 0, 1, 2, 3, 4, 5, 6 };

    var numQuery =
        from num in numbers
        where (num % 2) == 0
        select num;

    foreach (int num in numQuery)
    {
        Console.WriteLine("{0,1} ", num);
    }
}
```

Figura 5 – Exemplo de código com LINQ.

2.6.5. Lambda

O conceito das expressões lambda [36] nasceu com conceitos matemáticos chamados de cálculo lambda e que é usado para definir e estudar funções computáveis. Estas também são normalmente chamadas de “*inline functions*” por serem habitualmente funções pequenas e declaradas apenas numa pequena linha dentro de outras funções.

Em .NET, as expressões Lambda são métodos anónimos, o seu principal benefício é reduzirem o código das funções, que podem por vezes ser complexas.

Outra grande vantagem é que quando são declarados para execução directa, podem aceder às variáveis internas do método onde está o lambda como referência.

A sintaxe destas expressões é bastante simples e para criar uma destas expressões é necessário especificar os parâmetros de entrada, caso hajam, e por fim adicionar a expressão ou instrução.

```
customers.Where(c => c.City == "London");
```

Figura 6 – Exemplo de código Lambda

2.6.6. Entity framework

O *Entity Framework* é uma ferramenta da Microsoft, de mapeamento relacional entre objectos, que permite trabalhar com classes (entidades) que correspondem a tabelas de uma base de dados, tornando simples o acesso a estes dados e principalmente, eliminando a necessidade de escrever código de acesso às bases de dados (como por exemplo *SELECT*, *INSERT*, *UPDATE* e *DELETE*) (ver figura 7) na aplicação. Com o *Entity Framework* os desenvolvedores manipulam os dados através de classes que são mapeadas com as tabelas da base de dados. A comunicação do *Entity Framework* com a base de dados é feita através do *ADO.Net Provider*. Sendo assim, todos os comandos submetidos pelo *Entity Framework* são “traduzidos” para a linguagem da correspondente de base de dados, através do seu *provider*, gerando os comandos SQL mais adequados a cada operação com o máximo de desempenho [11].

Uma grande vantagem desta tecnologia é a possibilidade de utilizar LINQ para realizar todas as operações de manipulação da base de dados (*SELECTs*, *INSERTs*, *UPDATEs* e *DELETEs*).

```
using (var context = new BloggingContext())
{
    // Query for all blogs with names starting with B
    var blogs = from b in context.Blogs
                where b.Name.StartsWith("B")
                select b;

    // Query for the Blog named ADO.NET Blog
    var blog = context.Blogs
                  .Where(b => b.Name == "ADO.NET Blog")
                  .FirstOrDefault();
}
```

Figura 7 – Exemplo (Select) no *Entity Framework* [24]

2.6.7. Infragistics

Para a realização deste *site* foram utilizadas bibliotecas da *infragistics* [19] que é uma empresa que desenvolve componentes de interface para múltiplas plataformas. Neste caso utilizei os componentes na criação de tabelas e gráficos para *razor*. Um exemplo de uma grelha poderá ser encontrado em [25].

2.6.8. Web services

Os *web services* são utilizados para que diferentes aplicações possam comunicar e para que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

2.6.9. JSON

JSON "*JavaScript Object Notation*" foi criado por Douglas Crockford e é um formato utilizado para troca de dados. JSON tem-se difundido como uma alternativa ao uso de XML, devido à sua simplicidade e ao uso de convenções que são familiares em múltiplas linguagens de programação, como C, C++, C#, *Java*, *JavaScript*, entre outras [12].

Existe um crescente suporte para JSON através do uso de bibliotecas de terceiros.

O exemplo seguinte (figura 7) mostra um *array* com três objectos, nomeadamente os objectos com nomes “João”, “Maria” e “Pedro”, com as respectivas notas.

```
{ "Alunos" : [
    { "nome": "João", "notas": [ 8, 9, 7 ] },
    { "nome": "Maria", "notas": [ 8, 10, 7 ] },
    { "nome": "Pedro", "notas": [ 10, 10, 9 ] }
]
```

Figura 8 – Exemplo de conteúdo de ficheiro JSON

Se o código da imagem anterior fosse inserido no site <http://json2csharp.com/> as classes resultantes seriam (ver figura 8).

```
public class Aluno
{
    public string nome { get; set; }
    public List<int> notas { get; set; }
}

public class RootObject
{
    public List<Aluno> Alunos { get; set; }
}
```

Figura 9 – Exemplo de classe que recebe os dados do ficheiro JSON da figura 8

Por fim o JSON da figura 8 poderia ser passado para a classes apresentada na figura 9 utilizando o seguinte código (ver figura 10).

```
var outList = JsonConvert.DeserializeObject<RootObject>(json);
```

Figura 10 – Exemplo de código para obter o JSON para as classes

2.6.10. XML

Extensible Markup Language (XML) [37] é uma linguagem de marcação que define um conjunto de regras para a codificação de documentos.

É um formato de dados textual e embora o XML se foque na criação de documentos organizados, é amplamente utilizado para a representação de estruturas de dados arbitrários (ver figura 11), tais como aquelas utilizadas em *web services*.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<receita nome="pão" tempo_de_preparo="5 minutos" tempo_de_cozimento="1 hora">
  <titulo>Pão simples</titulo>
  <ingredientes>
    <ingrediente quantidade="3" unidade="xícaras">Farinha</ingrediente>
    <ingrediente quantidade="7" unidade="gramas">Fermento</ingrediente>
    <ingrediente quantidade="1,5" unidade="xícaras" estado="morna">Água</ingrediente>
    <ingrediente quantidade="1" unidade="colheres de chá">Sal</ingrediente>
  </ingredientes>
  <instrucoes>
    <passo>Misture todos os ingredientes, e dissolva bem.</passo>
    <passo>Cubra com um pano e deixe por uma hora em um local morno.</passo>
    <passo>Misture novamente, coloque numa bandeja e asse num forno.</passo>
  </instrucoes>
</receita>
```

Figura 11 - Exemplo de ficheiro em XML

2.6.11. CSS

Cascading Style Sheets (CSS) [38] é uma linguagem para criar folhas de estilo e é utilizada para definir a apresentação de documentos escritos numa linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.

Em vez de colocar a formatação dentro do documento, o desenvolvedor cria um *link* para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um *web site*. Quando quiser alterar a aparência do *web site* basta modificar apenas um ficheiro.

2.6.12. Ajax

Asynchronous Javascript and XML (AJAX) foi inicialmente desenvolvida pelo estudioso Jessé James Garret e mais tarde por diversas associações. Apesar do nome, a utilização de XML não é obrigatória (JSON é frequentemente utilizado) e as solicitações também não necessitam ser assíncronas [27].

```
$.ajax({
  type: 'GET',
  url: '@Url.Action("ChartAppCountry","SalesAndTrends" )',
  data: {
    App: appName,
  },
  dataType: 'html',
  success: function (result) {
    $('#PieChartCountry').html(result);
  },
  failure: function () { alert('@ViewResources.ERROnoajaxjavascript'); }
});
```

Figura 12 – Exemplo de código em AJAX

2.6.13. C#

C# ou *C Sharp* [39] é uma linguagem de programação orientada a objectos, desenvolvida pela Microsoft como parte da plataforma .NET. A sua sintaxe orientada a objectos foi baseada no C++ mas inclui muitas influências de outras linguagens de programação, como o Java.

2.6.14. Layout

Para poder manter uma aparência consistente em todas as páginas do *website*, o ASP.NET 2.0 introduziu o conceito de "*master-page*". *Razor* também suporta esse conceito com um recurso chamado de "*layouts*", que permite definir um template comum para o *site* e que faz herdar a sua aparência em todas as (vistas) páginas do site em que for aplicado [13].

2.6.15. Partialview

Uma *partial view* é apenas um *sub-view* que poderá ser incluída numa vista principal. Poderá ser reutilizada em varias páginas diferentes e mantém as vistas do site mais simples e divididas.

2.6.16. Membership

A classe de *Membership* é utilizada em aplicações ASP.NET para validar as credenciais dos utilizadores e controlar definições do utilizador, como as *passwords* e *e-mails*. A classe de *Membership* pode ser usada por si só ou em conjunto com *FormsAuthentication* para criar uma forma completa para autenticação de utilizadores de *web sites*.

2.6.17. Forms authentication

Forms authentication possibilita a validação de utilizadores nas *web applications*. A informação do utilizador é armazenada numa fonte de informação externa como uma base de dados, ou num ficheiro de configuração de uma aplicação. Uma vez que um utilizador esteja autenticado, *forms authentication* mantém informação da autenticação num *cookie* ou no URL para que o utilizador não tenha de fornecer as credenciais a cada *request* [14].

2.6.18. Metro Js

O Metro JS é um *plugin do JavaScript* para *jQuery*, desenvolvido para a fácil utilização das *interfaces* Metro na web [16].

2.6.19. Multiple select

Multiple select é um *plugin do jQuery* que permite criar uma *dropdown* de *checkboxes* e controlá-la através de diversos eventos [17].

2.6.20. Itunes Search API

A *Itunes Search API* é uma ferramenta para *sites* que procuram implementarem uma forma automática de gerar *links*. Esta ferramenta funciona submetendo uma *query* em formato de URL na Apple, a página que será carregada como resultado da *query* estará no formato JSON. A *Itunes Search API* não necessita de uma conta para ser utilizada e por isso depois de uma rápida revisão à documentação [3] poder-se-á começar a trabalhar com ela.

2.6.21. Chartboost Api

A *api* do *Chartboost* [4] permite obter informações das transacções feitas no *Chartboost* bem como fazer interacções através desta.

2.6.22. APNs

Apple Push Notification service é a peça central das notificações remotas. É um serviço robusto e altamente eficiente para propagar informações para o iOS e OSX. Cada aparelho estabelece uma credenciada e encriptada conexão IP com o serviço e recebe notificações sobre esta conexão. Se uma notificação para uma aplicação chegar quando essa aplicação não estiver a correr, o aparelho alerta o utilizador que a aplicação tem uma notificação à sua espera.

Desenvolvedores de *software* (“*providers*”) originam a notificação no seu servidor. O *provider* conecta-se com a APNs através de um canal seguro e persistente, enquanto controla dados que chegam das suas aplicações do cliente. Quando novos dados para uma aplicação chegam, o *provider* prepara e envia a notificação através do canal para as APNs, que por sua vez envia a notificação para o *Iphone* alvo [28].

2.6.23. PushSharp

É uma biblioteca para enviar *push notifications* para iOS (*iPhone/iPad* APNS), OSX (APNS 10.8+) Android (C2DM and GCM - *Google Cloud Message*), *Chrome* (GCM), *Windows Phone*, Windows 8, Blackberry (PAP), entre outros [15].

2.6.24. Regex

Uma expressão regular (*regex regular expression*) fornece uma forma concisa e flexível de identificar cadeias de caracteres de interesse, como caracteres particulares, palavras ou padrões de caracteres (ver figura 13).



Figura 13 – Exemplo de uma expressão regular [26] para números entre 200 e 249

2.6.25. DataAnnotations

DataAnnotations é uma biblioteca do .NET. A *DataAnnotations* permite configurar os elementos de uma classe com atributos. Para isso, *DataAnnotations* contém atributos de validação para fazer cumprir as regras de validação e atributos de *interface* para especificar como os elementos da classe são mostrados. Um exemplo de um atributo de validação poderá

ser o *Required* que é utilizado para dizer que um valor tem de ser especificado. Um exemplo de um atributo de *display* é *Display* que é utilizado para definir a *string* que será mostrada na *interface* de utilizador neste caso na página *web*.

Em seguida é mostrado um exemplo dos *DataAnnotations* (figura 14) e da validação que é feita (figura 15).

```
//adicionar a biblioteca
using System.ComponentModel.DataAnnotations;
//tamanho maximo da string 260 utilizando os recursos para que a mensagem esteja na linguagem correcta
[StringLength(260, ErrorMessageResourceType = typeof(ValidationMessages), ErrorMessageResourceName = "Maxlength260")]
//string mostrada para o LinkToStore tambem utilizando resources
[Display(Name = "LinkToStore", ResourceType = typeof(ViewResources))]
public string LinkToStore { get; set; }
```

Figura 14 – Exemplo de *DataAnnotations* do lado do servidor

```
//bibliotecas de javascript a adicionar
<script src="@Url.Content("~/Scripts/jquery.validate.min.js")"
type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>
//vai criar a label para SuportUrl
@Html.LabelFor(model => model.SuportUrl)
//vai criar a textbox para SuportUrl
@Html.TextBoxFor(m => m.SuportUrl)
//vai ser mostrada a mensagem de validação caso não passe na validação
@Html.ValidationMessageFor(model => model.SuportUrl)
```

Figura 15 – Exemplo de *DataAnnotations* do lado do cliente

2.7. Páginas do IBO

O site IBO da empresa Kamae é composto por várias páginas cada uma com as suas características e funcionalidades. Nos subcapítulos seguintes é explicado o que pode ser feito em cada umas das seguintes páginas, página de *login*, página de registo, página principal, página das *apps*, página de *more screens*, página de *nag screens*, Página de *push notifications*, página de *sales and trends*, página de *outsource* e página de opções.

2.7.1. Página de Login

A figura seguinte (figura 16) representa a página de *login* do site IBO onde o utilizador poderá iniciar sessão.

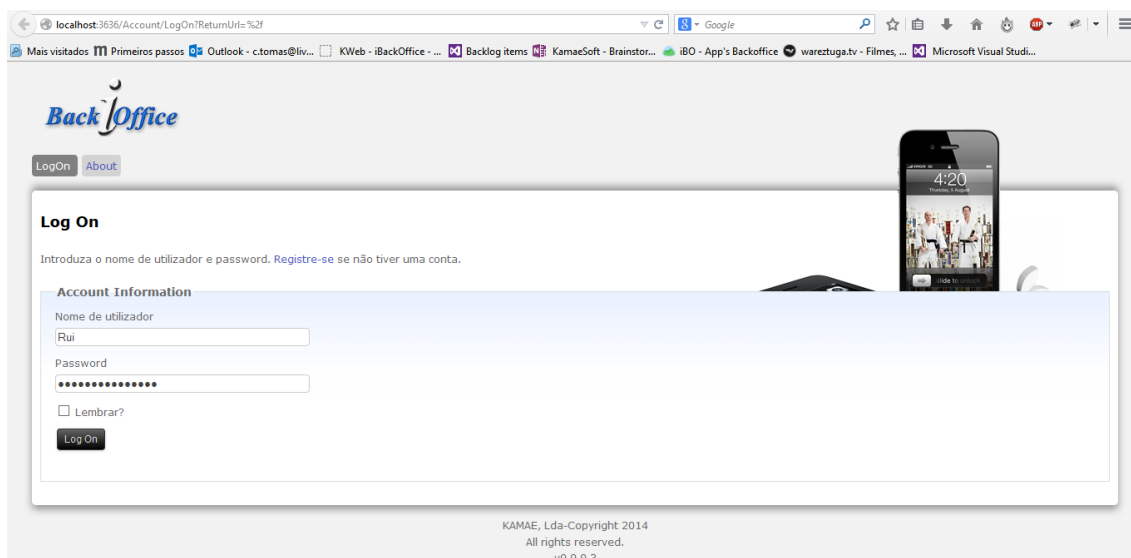


Figura 16 - Página de *login*

Esta página será mostrada caso o utilizador escreva o seu URL no *browser*. Na situação do utilizador tentar entrar numa página que necessite de ter sessão iniciada e esta não o esteja então o utilizador será redireccionado para a página de *login*, em caso de *login* bem-sucedido o utilizador será redireccionado para a *home page* por defeito ou para a página em que tenha tentado entrar anteriormente e tenha sido redireccionado para a página de *login*. No caso de o *login* ser mal sucedido o utilizador vai ser mantido na página de *login*. Desta página e sem sessão iniciada o utilizador vai poder navegar para a página de registo ou para a página *about*.

A sessão é manuseada utilizando as classes *Membership* e *FormsAuthentication*.

O utilizador poderá registar-se inserindo o nome de utilizador e *password*.

2.7.2. Página de registo

A imagem seguinte (figura 17) representa a página de registo do site IBO, onde o utilizador poderá registar-se inserindo nome de utilizador, mail, *password* e confirmação de *password*.

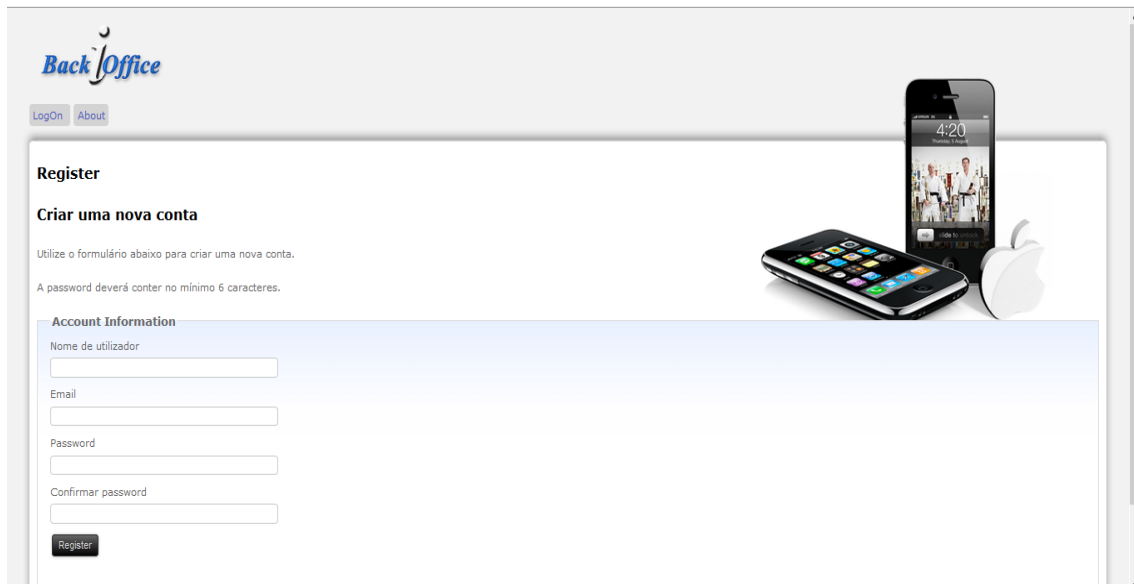
The image shows a web browser window displaying the 'BackOffice' registration page. At the top left is the 'BackOffice' logo. Below it are 'LogOn' and 'About' links. The main heading is 'Register', followed by the sub-heading 'Criar uma nova conta'. Below this, there are two lines of instructional text: 'Utilize o formulário abaixo para criar uma nova conta.' and 'A password deverá conter no mínimo 6 caracteres.' To the right of the text is an image of an iPhone and an Apple logo. Below the text is a form titled 'Account Information' with four input fields: 'Nome de utilizador', 'Email', 'Password', and 'Confirmar password'. At the bottom of the form is a 'Register' button.

Figura 17 - Página de registo

Desta página o utilizador poderá navegar para as páginas de *login* e *about*.

2.7.3. Página principal

A figura seguinte (figura 18) representa a página principal da aplicação (figura 16).



Figura 18 - Página principal (*Home page*)

Através desta página, o utilizador poderá navegar para qualquer uma das outras páginas clicando num dos botões do menu que se encontra no topo do site (*Home, Apps, More screens, Nag screens, Push notifications, Sales and Trends, OutSource e Options*). O utilizador poderá ainda fazer *log out*, clicando no *link Log Off* no Topo direito da página. Os dois componentes *log off* e *menu* fazem parte do *layout* e são comuns às oito páginas que podem ser acedidas através do *menu*.

Nesta página existem *tiles*, feitos utilizando o *plugin Metro Js* [16]. Estas *tiles* permitem ao utilizador verificar rapidamente quantas aplicações existem, lucros obtidos, o rácio para cada mil downloads (lucro obtido para cada mil downloads), lucro obtido com a venda de *inapps*, numero total de *downloads* feitos, o País em que as *apps* tiveram mais *downloads*, a aplicação com mais *downloads*, a pior aplicação em número de *downloads*, o número de *downloads* do dia anterior, número de cliques em *nag screens* hoje, dia da última actualização, e lucros em publicidade.

2.7.4. Página das apps

A imagem seguinte (figura 19) mostra a página das aplicações do *site IBO*.

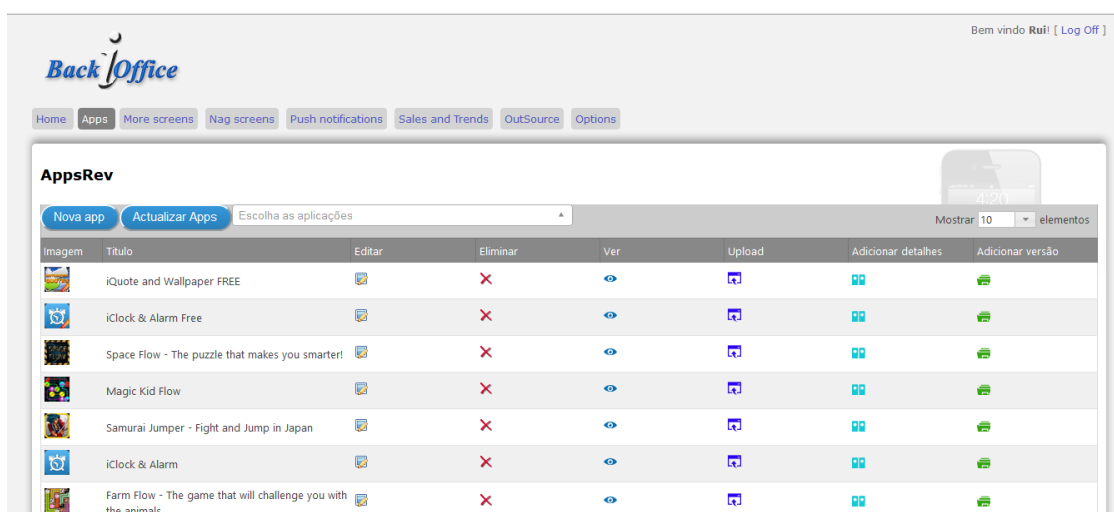


Figura 19 - Página das aplicações do site IBO.

As aplicações estão representadas na forma de grelha (colunas: “Imagem”, “Titulo”, “Editar”, “Eliminar”, “Ver”, “*Upload*”, “Adicionar detalhes” e “Adicionar versão”). Esta foi a forma encontrada para dispor o máximo de informação possível e ao mesmo tempo possibilitar encontrar uma aplicação e interagir com esta rapidamente.

No caso de o utilizador ter muitas aplicações, poderá fazer a pesquisa utilizando uma *dropdown* de *check boxes* que se encontra no topo da grelha (ver figura 20) onde não só poderá fazer pesquisa mas também seleccionar as várias aplicações com que pretende interagir.

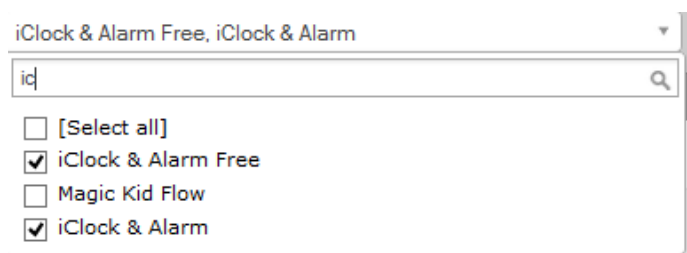


Figura 20 – Exemplo de *dropdown* de *check boxes*

O utilizador poderá adicionar uma nova aplicação clicando no botão *Nova app* no topo da grelha fazendo aparecer uma *dialog box* onde o utilizador poderá adicionar o Título, o *Bundle Id*, o *Apple Identifier*, o SKU e seleccionar as categorias (ver figura 21) da aplicação que pretende inserir.

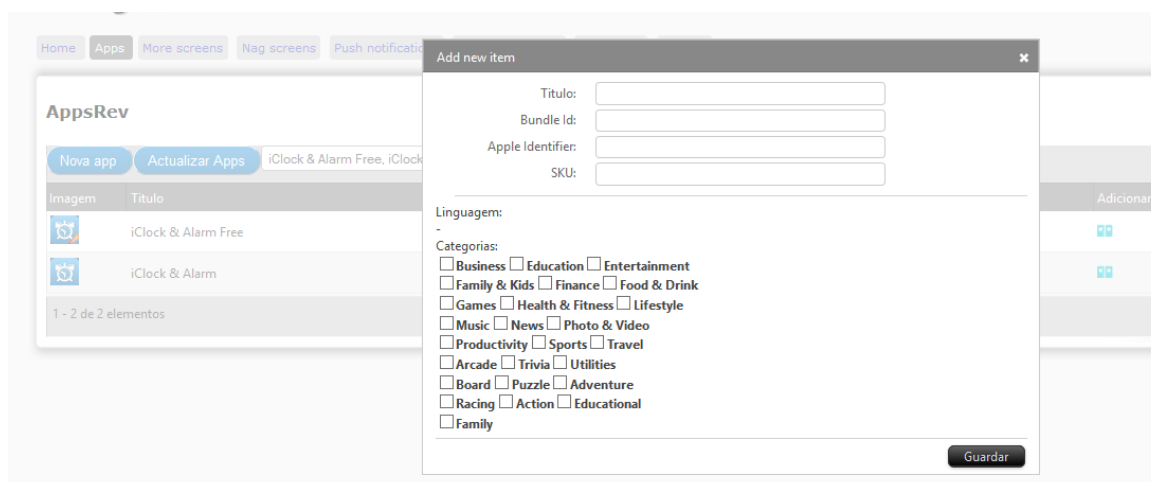


Figura 21 - Formulário para inserir uma aplicação no site IBO

O utilizador poderá actualizar as aplicações pelo *Itunes Search API*, clicando no botão *Actualizar Apps* que se encontra no topo da grelha com das aplicações. Será feito, um pedido à API das aplicações de determinado utilizador que devolverá um ficheiro no formato JSON [5], em seguida através dos dados do ficheiro adiciona novas aplicações caso existam e actualiza as aplicações já introduzidas caso tenha existido alguma actualização.












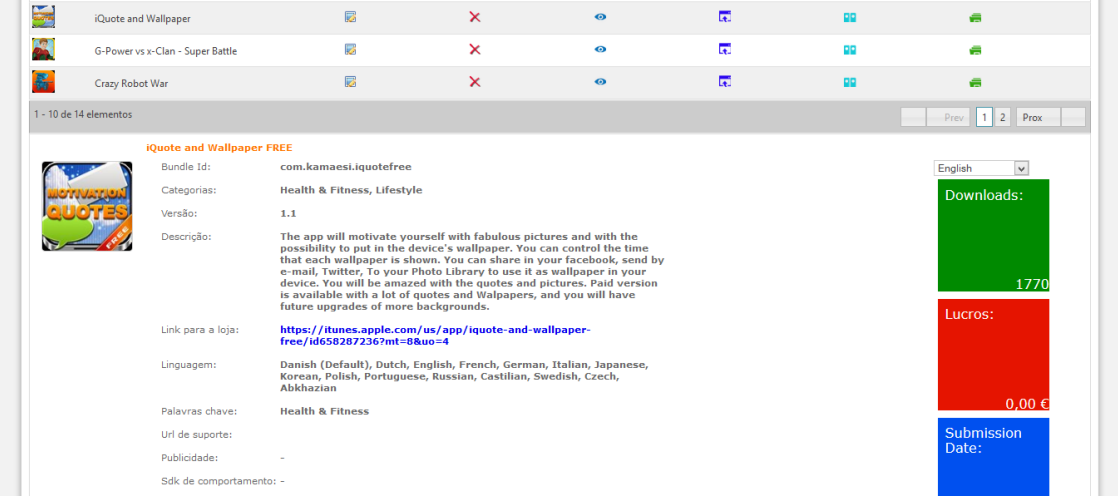
Imagem	Título	Editar	Eliminar	Ver	Upload	Adicionar detalhes	Adicionar versão
	iQuote and Wallpaper FREE						
	iClock & Alarm Free						

Figura 22 - Parte da grelha das aplicações.

O utilizador poderá editar uma aplicação clicando na imagem da coluna “Editar” da aplicação para a qual pretende fazer alterações, o que fará aparecer uma *dialog box* onde o utilizador poderá editar o “Título”, “*Bundle Id*”, “*Applle Identifier*”, “Sku” e marcar/desmarcar as categorias.

O utilizador poderá eliminar as aplicações clicando na imagem da coluna “Eliminar” da grelha. Será apresentada uma mensagem para o utilizador confirmar se realmente pretende apagar a aplicação e caso o utilizador diga que sim o sistema irá eliminar a aplicação se não tiver sido actualizada pelo *Itunes Search API* e se não tiver dados financeiros, caso contrario, a aplicação passa para um estado de suspenso, continuando a existir mas não sendo possível visualizar os seus dados.

O utilizador poderá ver a aplicação em maior detalhe clicando na imagem da coluna “Ver” que fará com que por baixo da grelha das aplicações surjam mais informações sobre a aplicação (ver figura 23 e 24).



1 - 10 de 14 elementos

iQuote and Wallpaper FREE

Bundle Id: com.kamaesi.iquoteefree

Categorias: Health & Fitness, Lifestyle

Versão: 1.1

Descrição: The app will motivate yourself with fabulous pictures and with the possibility to put in the device's wallpaper. You can control the time that each wallpaper is shown. You can share in your facebook, send by e-mail, Twitter. To your Photo Library to use it as wallpaper in your device. You will be amazed with the quotes and pictures. Paid version is available with a lot of quotes and Walpapers, and you will have future upgrades of more backgrounds.

Link para a loja: <https://itunes.apple.com/us/app/iquote-and-wallpaper-free/id658287236?mt=8&uo=4>

Linguagem: Danish (Default), Dutch, English, French, German, Italian, Japanese, Korean, Polish, Portuguese, Russian, Castilian, Swedish, Czech, Abkhazian

Palavras chave: Health & Fitness

Uri de suporte: -

Publicidade: -

Sdk de comportamento: -

English

Downloads: 1770

Lucros: 0,00 €

Submission Date:

Figura 23 - Ver aplicação no *site* IBO (parte 1)

Nesta parte do *site* poderão ser vistas informações relativas á última versão da *app*, será possível ainda escolher a língua e visualizar a informação correspondente (não quer dizer que a informação representada se encontre na língua escolhida, mas sim que, para essa língua estes dados estão representadas desta forma na *app store*, caso estes tenham sido obtidos da *Itunes Search API*, ou porque o utilizador escolheu inserir os dados desta forma directamente na aplicação).

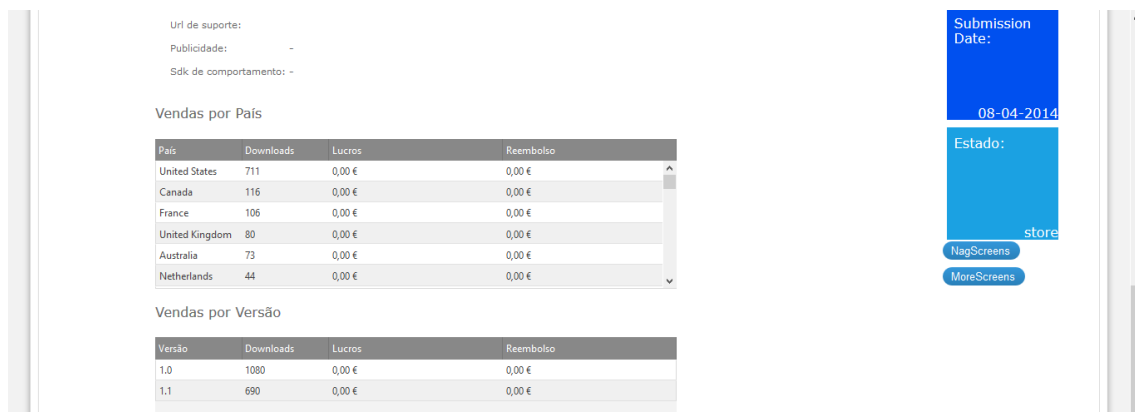


Figura 24 - Ver aplicação no site IBO (parte 2)

Os dados que podem ser vistos na página de apps quando se clica em ver (figuras 21 e 22), são “Titulo”, “*Bundle Id*”, “Categorias”, “Versão”, “Descrição”, “Link para a loja”, “Linguagem” (a linguagem em que se encontra disponível), “Palavras chave”, “Url de suporte”, “Publicidade” (que publicidade utiliza o *Chatboost* poderá ser um exemplo), “Sdk de comportamento” (“*Flurry*” poderá ser um exemplo). Depois poderão ser vistas tabelas com informação dos *downloads*, vendas por país, vendas por versão, e outsource. Do lado direito, o utilizador poderá visualizar uma *dropdown* com a linguagem onde pode seleccionar uma linguagem e os dados vistos mudarão conforme esta selecção, e em seguida 4 *tiles* com informação do numero de *downloads*, lucros, data de submissão, e estado (*store* – a aplicação esta na loja, sem estado - inserida pelo utilizador e *suspended* - caso tenha sido eliminada). Em seguida existem dois botões para o utilizador ir para as páginas de *nag* e *more screens*.

Clicando no botão de *upload*, será mostrado ao utilizados uma *dialog box* (ver figura 25) onde este poderá fazer o *upload* do ficheiro, necessário para enviar *push notifications* para a aplicação e poderá inserir uma *password* caso haja uma, inserir um ficheiro no formato “.pem” ou “.p12” e seleccionar a *checkbox* “produção”, no caso das notificações serem para enviar para os utilizadores finais da aplicação. No caso em que estas sejam apenas para teste, ou seja para enviar notificações para *sand boxe*(a *sand box* quer dizer que apenas são enviadas notificações para *Iphones* de teste) deixar está desmarcada. Depois basta ao utilizador clicar em *upload* e caso o ficheiro esteja no formato correcto será guardado no servidor.

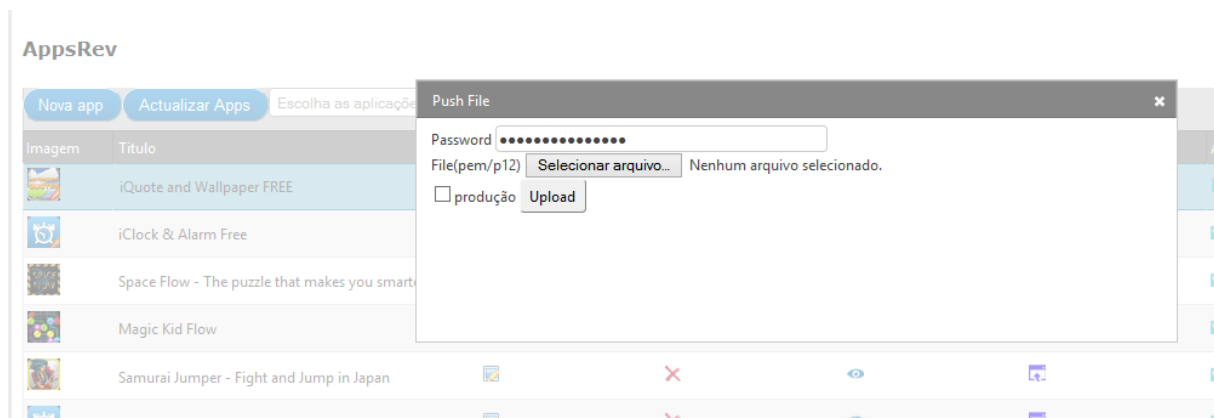


Figura 25 - Formulário para fazer *upload* de um ficheiro para *push notifications* no site IBO.

O utilizador poderá editar uma aplicação clicando na imagem da coluna “Adicionar detalhes” o que fará surgir uma *dialog box* (ver figura 26) onde o utilizador poderá adicionar ou editar os detalhes já existentes. Para editar detalhes já existentes o utilizador poderá marcar a *check box* “Editar” e na *dropdown* irão aparecer as linguagens para as quais já existem detalhes e em cada um dos campos os respectivos dados preenchidos. Caso a *check box* esteja desmarcada aparecerão as restantes linguagens para as quais não existem detalhes. O utilizador poderá seleccionar a versão para a qual quer adicionar detalhes, seleccionar a linguagem, adicionar uma descrição, adicionar um *link* para a loja, palavras chave, o *URL de suporte*, seleccionar a publicidade que utiliza e o Sdk de comportamento, e por fim clicar em guardar para guardar as alterações.

Figura 26 - Formulário para inserir detalhes de uma aplicação no *site* IBO.

O utilizador poderá adicionar uma nova versão de uma aplicação clicando na imagem da coluna adicionar versão o que fará surgir uma *dialog box* (a versão deverá estar no formato número, ponto, número, ponto, (...)) devendo terminar sempre com número).

2.7.5. Página de *More screens*

A página seguinte (Figura 27) representa a página dos *more screens* do *site* IBO.

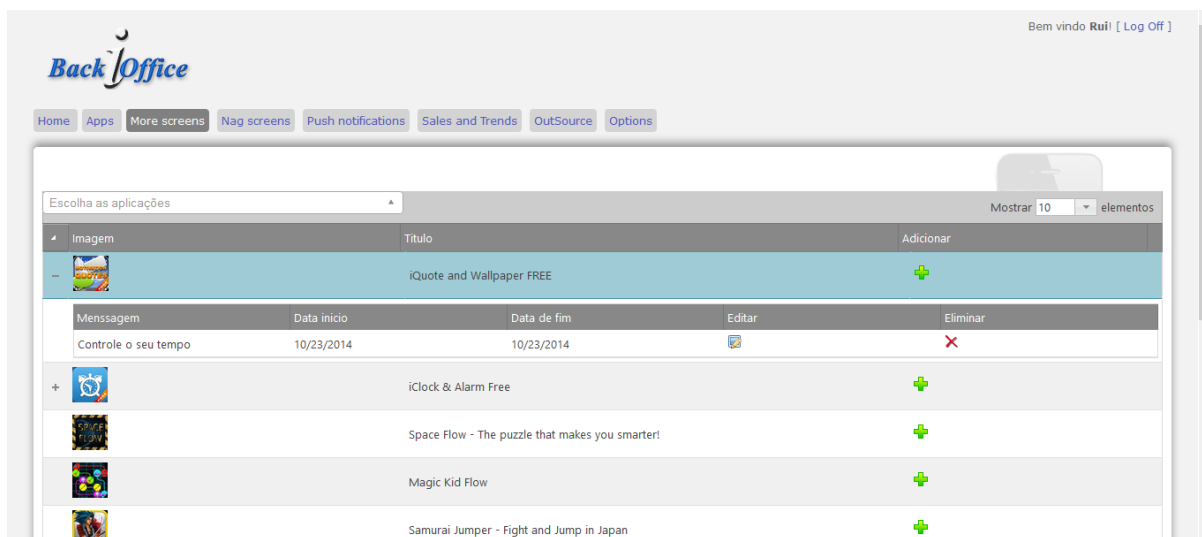


Figura 27 - Pagina *More screens* do site *IBO*.

Esta página é em tudo muito idêntica a página das aplicações (*Apps*) com as aplicações dispostas em grelha com a imagem da aplicação, o título e um botão para adicionar mais *more screens*. As aplicações que têm *more screens* têm um sinal de “+” antes. Este sinal, poderá ser clicado pelo utilizador, o que fará expandir uma subgrelha onde podem ser visualizados os *more screens* disponíveis (mensagem, data início, data fim, imagem para editar e imagem para eliminar).

Clicando na imagem de “Adicionar” surgirá uma *dialog box* onde o utilizador poderá seleccionar uma imagem, escolher uma linguagem, adicionar o título, descrição, fornecedor, data início e data de fim. A data de fim nunca poderá ser inferior a data de início. Clicando em guardar o site IBO irá guardar os dados ou apresentará mensagens a avisar, caso estes valores não respeitem certas restrições de inserção (validações *DataAnnotations* por exemplo o utilizador será obrigado a preencher o “Título”).

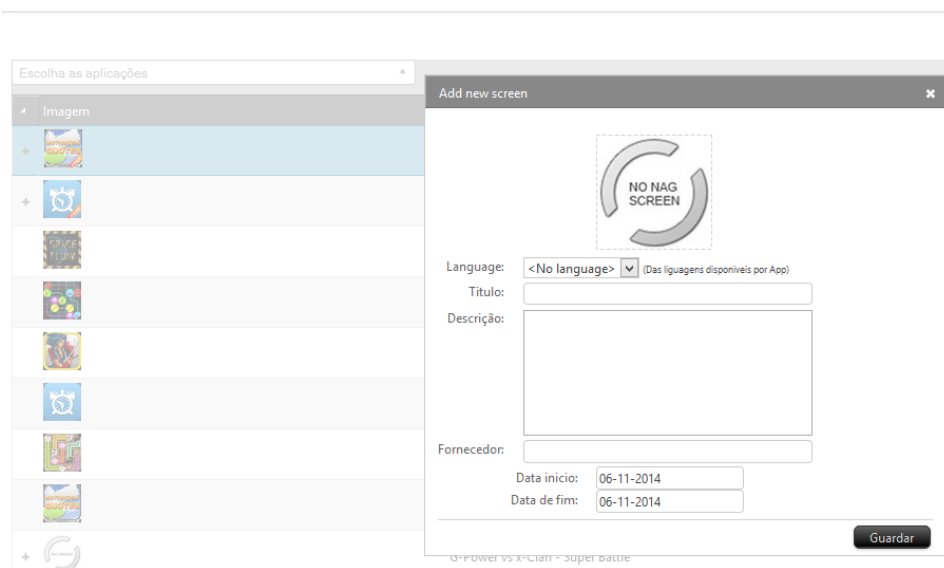


Figura 28 - Formulário para inserir *more screens* no site *IBO*

Clicando na imagem de eliminar o utilizador poderá eliminar o *more screen*.

Clicando na imagem de editar surgirá uma *dialog box* onde o utilizador poderá alterar o *more screen* seleccionando uma imagem, escolher uma linguagem, adicionar o título, descrição, fornecedor, data início, data de fim. A data de fim nunca poderá ser inferior a data de início. Clicando em guardar o *site* IBO irá guardar as informações, caso estes valores respeitem certas restrições de inserção (validações *DataAnnotations*).

2.7.6. Página de nag screens

A página seguinte (ver figura 29) representa a página dos *nag screens* do *site* IBO.

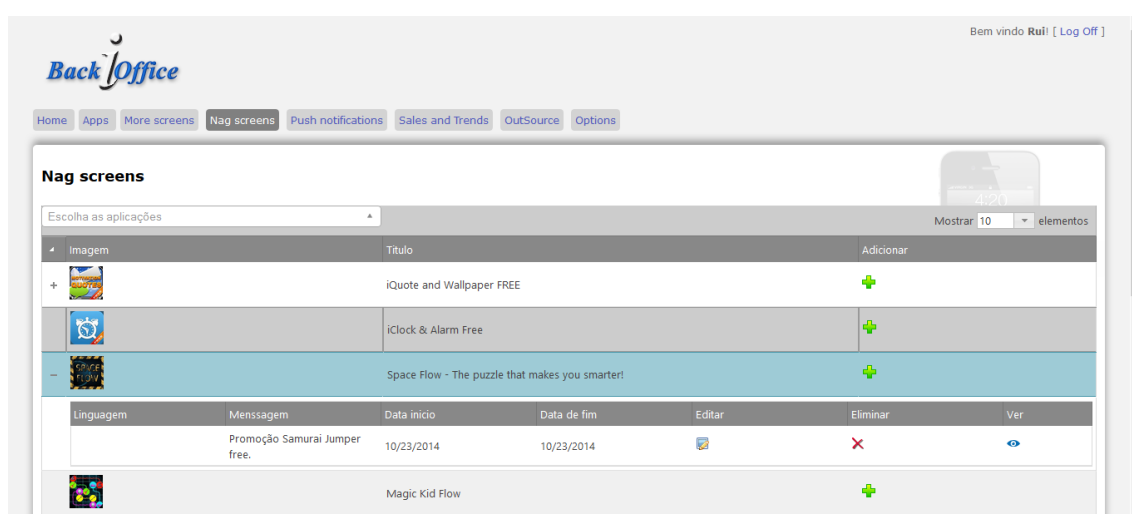


Figura 29 - Página de *nag screens* do *site* IBO

Esta página é em tudo muito idêntica a página dos *more screens* com as aplicações dispostas em grelha com a imagem da aplicação, o título e um botão para adicionar mais *nag screens*. As aplicações que possuem *nag screens* têm um sinal de “+” junto à imagem. Este sinal poderá ser clicado para o utilizador expandir uma subgrelha, onde poderá ver os *nag screens* disponíveis (mensagem, data início, data fim, imagem para editar, imagem para eliminar e imagem para visualizar).

Clicando na imagem de “Adicionar” ou na imagem de “Editar” surgirá uma *dialog box* (ver figura 30) onde o utilizador poderá seleccionar se quer apenas introduzir texto seleccionando a opção texto ou imagem onde poderá adicionar imagem e texto. Os campos para preencher ou editar são os seguintes: seleccionar imagem, adicionar o texto, escolher uma linguagem, se está activo, data início, data de fim e onde o *nag screen* será mostrado “Show on”. A data de fim nunca poderá ser inferior a data de início. Clicando em guardar, o *site* IBO irá guardar o *nag screen*. O utilizador poderá ainda eliminar o *nag screen* clicando na Imagem de “Eliminar”.

The screenshot shows the 'BackOffice' interface with a navigation bar containing links: Home, Apps, More screens, Nag screens, Push notifications, Sales and Trends, OutSource, and Options. The 'Nag screens' section is active, displaying a list of application icons on the left and a configuration modal on the right. The modal, titled 'Add nag screen', has two tabs: 'Text' and 'Image', with 'Image' selected. It features a preview of a circular 'NO NAG SCREEN' icon. Below the preview are fields for 'Texto:', 'Linguagem:' (set to '<Any>'), 'Está activo:' (checkbox), 'Data inicio:' (06-11-2014), 'Data de fim:' (06-11-2014), and 'Show On:' (init). A 'Guardar' button is at the bottom right.

Figura 30 - Formulário para inserir *nag screens* no site IBO

2.7.7. Página *Push notifications*

A figura seguinte (figura 31) representa a página de *push notifications* que tem como função o envio das *push notifications* para as aplicações. Para isso o utilizador do *site* IBO necessita de seleccionar uma ou mais aplicações, ou então seleccionar a *checkbox* para todas, o que vai fazer com que todas as aplicações fiquem seleccionadas. Para seleccionar uma aplicação basta clicar na mesma e o utilizador poderá seleccionar a linguagem para a qual quer enviar a mensagem, o que irá fazer com que a mensagem só seja recebida por utilizadores com o *iPhone* naquela linguagem, mas também há a possibilidade de seleccionar “Any” para que a mensagem seja recebida por todos os utilizadores independente da linguagem no seu *iPhone*, preenchendo o campo da “Mensagem” e em seguida clicar em enviar para enviar a notificação para o utilizador. A opção de utilização de *sandbox*, quando seleccionada, quer dizer que será utilizada para testes ou seja para enviar *push notifications* para aplicações em teste que estejam nos *IPhones* designados para isso. Quando a aplicação é posta em produção e o utilizador deseja enviar *push notifications* para aplicações retiradas da *store* será preciso desmarcar esta *checkbox*.

The screenshot shows the 'BackOffice' interface with the 'Push notifications' section active. It includes a language dropdown set to 'Abkhazian', a message input field, and checkboxes for 'Para todos' and 'Utilizar SandBox'. Below these are buttons for 'Enviar', 'Mostra histórico', and 'Push avançada'. A list of applications is shown with pagination controls. The first application is 'Ants Crazy Race' with a description and bundle ID. The second is 'IKL - Time sheet' with a detailed description.

Figura 31 - Página de *push notifications* do *site* IBO

O utilizador poderá pesquisar as aplicações na *text box* com o ícone de lupa. Poderá ainda seleccionar o número de aplicações visíveis por página.

O utilizador poderá enviar *push notifications* mais personalizadas clicando em *push* avançadas (ver figura 32) onde poderá fazer escolhas mais dinamicamente.

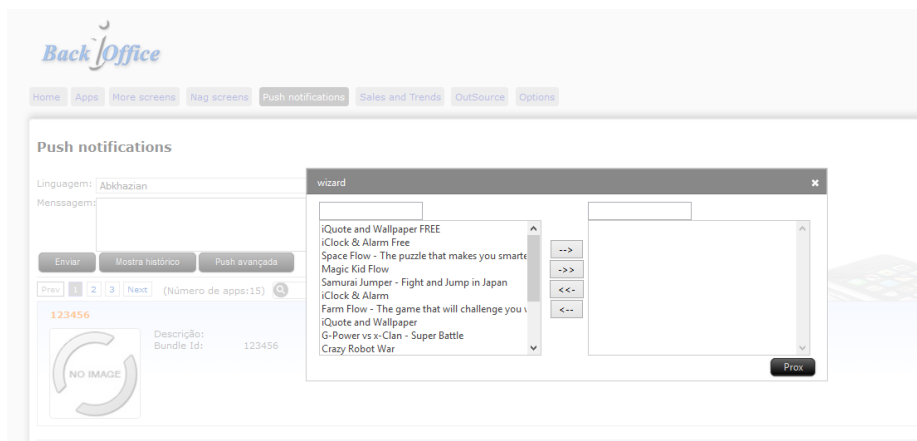


Figura 32 - Formulário para enviar *push notifications*.

2.7.8. Página de *sales and trends*

As imagens seguintes (figura 33, 34 e 35) representam a página de “*Sales and trends*” onde o utilizador poderá consultar os seus dados financeiros.

Na parte superior desta página o utilizador poderá ver *tiles* com informações do número de *downloads*, lucros das aplicações e reembolsos. O botão de “*update*” ao ser pressionado faz que os valores sejam actualizados do *itunes connect* (esta funcionalidade poderá demorar algum tempo, caso seja a primeira actualização, e dependendo da quantidade de aplicações e dias que serão actualizados). Ao lado direito do botão de *update* encontra-se a *dropdown* de *check boxes* que como em outras páginas contém os nomes das aplicações, quando utilizador seleccionar uma ou mais aplicações. A grelha e gráficos serão actualizados para conter apenas informações relativas a essas aplicações. A grelha encontra-se disposta por nome da aplicação, *downloads*, lucros e reembolsos.

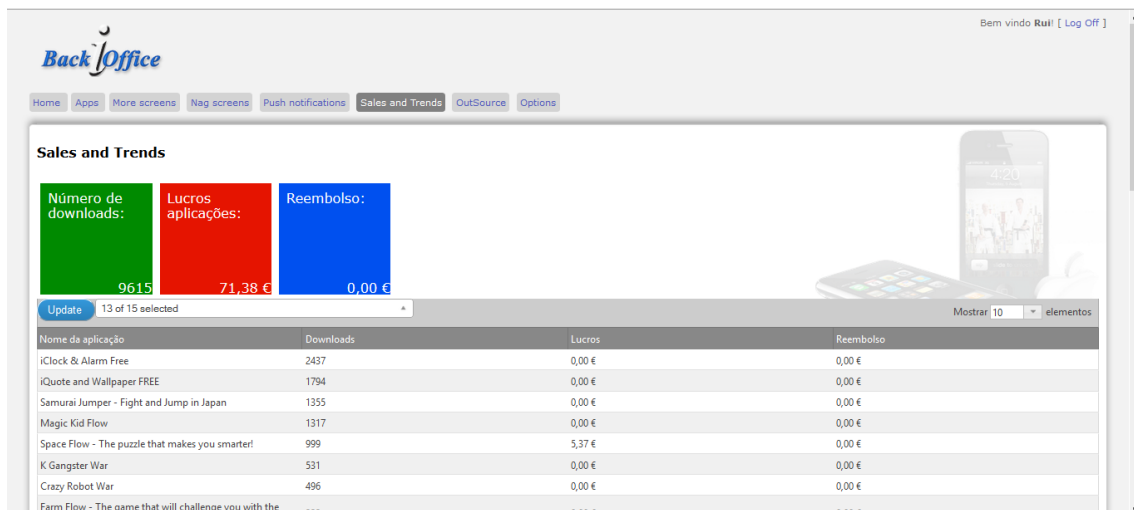


Figura 33- Página *Sales and Trends* do site *IBO* (parte 1)

Por baixo da grelha o utilizador poderá encontrar gráficos relativos às aplicações. O primeiro gráfico de barras à direita, representa o número de *downloads* por aplicação. No segundo, os lucros obtidos por aplicação e no terceiro, um gráfico circular que representa o número de *downloads* por país. Os valores poderão ser vistos passando com o rato por cima do respectivo país.

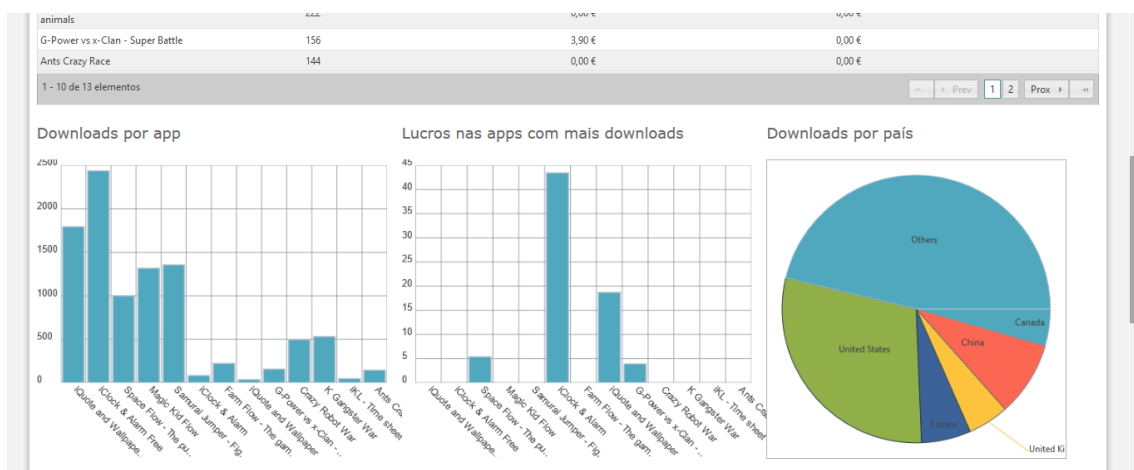


Figura 34 - Página *Sales and Trends* do site *IBO* (parte 2)

Existe uma segunda linha de gráficos da qual o primeiro gráfico de barras à direita representa o número de *downloads* por versão de uma determinada aplicação, seleccionada na *dropdown* em cima do gráfico. O gráfico circular representa o número de *downloads* por país de uma determinada aplicação. Os valores podem ser vistos passando com o rato por cima do respectivo país e o último gráfico representa o número total de *downloads* por dia.

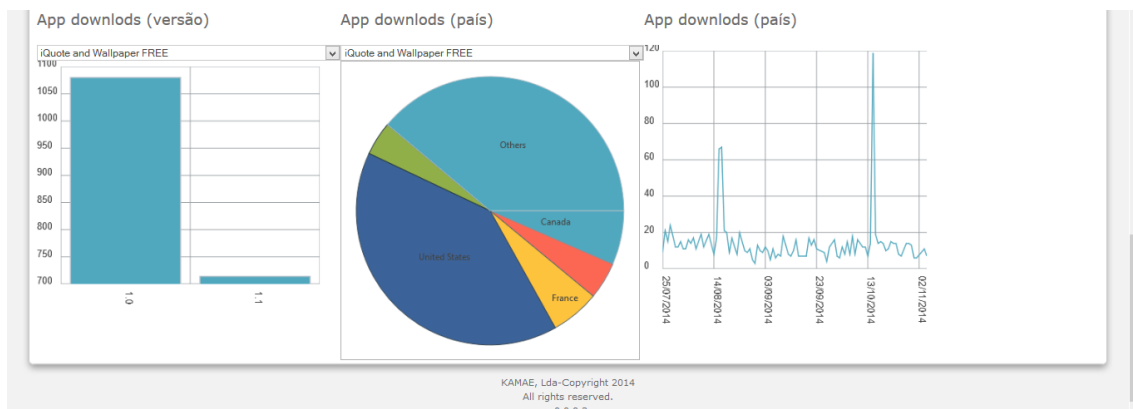


Figura 35 - Página *sales and trends* do site IBO (parte 3)

2.7.9. Página de *outsources*

A página seguinte (figura 36) representa a página dos *outsources* nesta página pretende-se guardar informação das pessoas externas, que trabalharam com as aplicações, como os *designers*, os programadores e outros. À semelhança de outras páginas, é mostrada uma grelha com as aplicações, no caso de uma aplicação ter *outsources* esta poderá ser expandida surgindo uma subgrelha onde o utilizador poderá ver o nome do *outsources*, função, *email*, contacto telefónico e as datas de início e fim de trabalhos.

BackOffice

Home Apps More screens Nag screens Push notifications Sales and Trends OutSource Options

OutSource

Escolha as aplicações

Mostrar 10 elementos

Imagem	Título	Adicionar
	iQuote and Wallpaper FREE	
	iClock & Alarm Free	
	Space Flow - The puzzle that makes you smarter!	
	Magic Kid Flow	
	Samurai Jumper - Fight and Jump in Japan	
	iClock & Alarm	

Nome	Função	Email	Contacto telefonico	Data contrato	Data inicio	Data fim	Editar	Eliminar
Jorge Andre	Designer	algo@hotmail.com	919876543	18/09/14	18/09/14	30/09/14		

Figura 36 - Página de *outsources* do site IBO

Clicando na imagem (“+”) de “Adicionar” ou na imagem de “Editar” da grelha, surgirá uma *dialog box* onde o utilizador poderá introduzir o nome, país, *link*, classificação, email, número de telefone, função, data de contrato, data início e data de fim. A data de fim nunca poderá ser inferior a data de início. Clicando em guardar o IBO guarda o *outsources*. O utilizador poderá ainda eliminar o *outsources* clicando no botão de “Eliminar”.

Out details

Nome:

País:

Link:

Classificação:

Email:

Número de telefone:

Função:

Data contrato:

Data inicio:

Data fim:

Função	Email
Designer	algo@f

Adicio

Data

30/09

Samurai Jumper - Fight and Jump in Japan

Figura 37 - Formulário para inserir um *outsorce* no *site* IBO

2.7.10. Página de opções

A imagem seguinte (figura 38) representa a página das opções. Esta página tem como objectivo guardar as informações necessárias para que o IBO possa interagir com o *chartboost* e o *apple connect* de forma a poder obter as informações financeiras.

Bem vindo Rui! [Log Off]

BackOffice

Home Apps More screens Nag screens Push notifications Sales and Trends OutSource Options

Options

Clicando no botão seguinte aparece um pop up que lhe permite configurar as suas opções de ligação ao iTunes Connect.

Clicando no botão seguinte aparece um pop up que lhe permite configurar as suas opções de ligação ao CharBoost.

KAMAE, Lda-Copyright 2014
All rights reserved.
v0.0.0.3

Figura 38 - Página de opções do *site* IBO

Quando o utilizador clicar no primeiro botão de “Update” aparecerá uma *dialog box* onde este poderá introduzir as suas credenciais para entrar aceder a dados do *itunes connect*. Aqui, deverá introduzir o nome e *password* de acesso ao *itunes connect* bem como o id do vendedor, necessários para adquirir os ficheiros com informação sobre os *downloads*.

O utilizador deverá inserir o *software developer* para poder obter dados das aplicações através da *apple search api*.

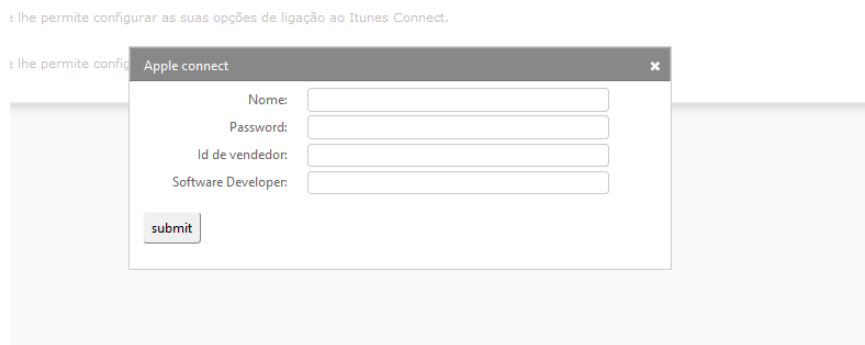


Figura 39 - Formulário para introduzir credenciais do *Apple connect* do *site* IBO

Quando o utilizador ao clicar no segundo botão de “Update”, aparecerá uma *dialog box* onde este poderá introduzir as suas credenciais para aceder a dados do *chartboost*: Aqui deverá introduzir o *User ID* e *User Signature* (estes poderão ser obtidos em [4]) para que se possa utilizar a *Web Api* do *chartboost* e recolher os dados financeiros desta.

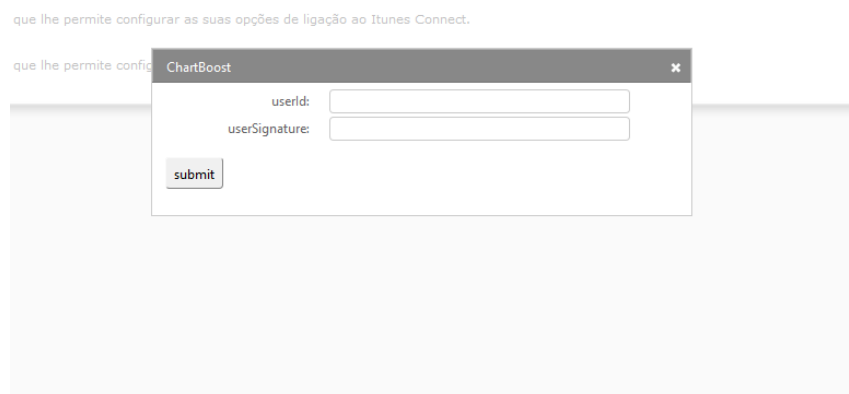


Figura 40 - Formulário para inserir credenciais no *chartboost* do *site* IBO

2.8. Integração com as aplicações

Foi ainda necessário, por fim, integrar as funcionalidades com as aplicações para isso em conjunto com o membro da equipa que desenvolveu as aplicações para *iphone* foi feita uma mostra de conceito e posteriormente a integração. As imagens seguintes mostram um *nag screen* (figura 41) e uma *push notification* (figura 42) numa aplicação da Kamae.

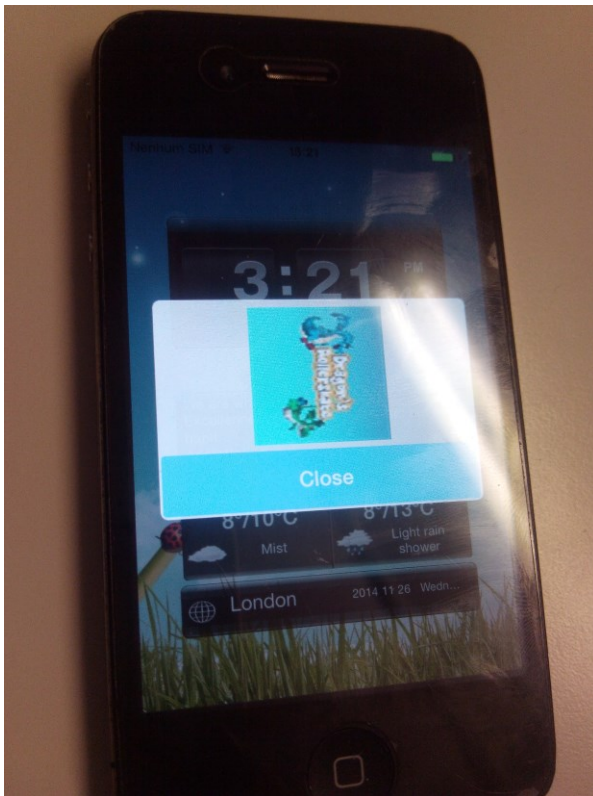


Figura 41 - Nag screen iPhone

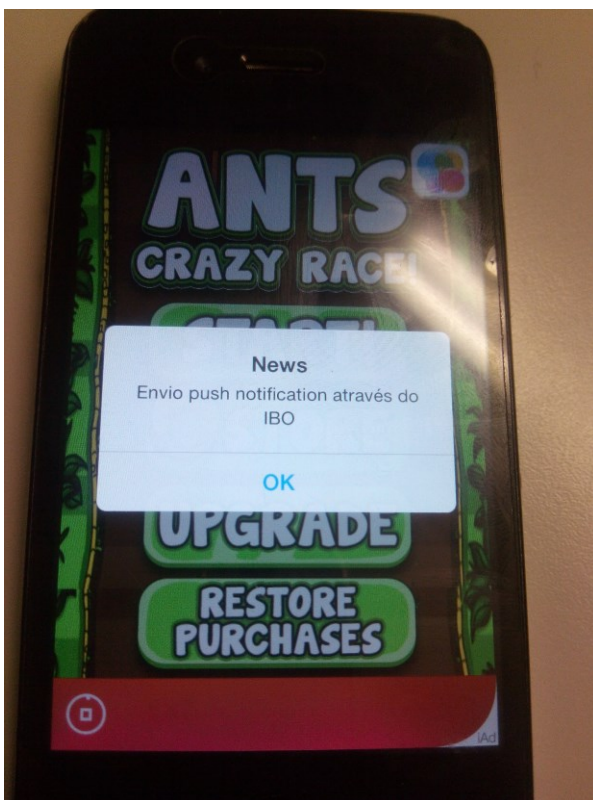


Figura 42 – Imagem de push notification

O utilizador poderá clicar no *nag screen* e ir para a página da aplicação correspondente.

Em termos de *push notifications*sum *Iphone* com uma das aplicações da empresa envia um *token* para o servidor. Do lado do servidor quando é recebido o *token* para as *push notifications* é verificado se este existe, e se não existir, se o telemóvel correspondente já foi registado para aquela aplicação se sim o *token* é substituído.

O *nag screen* poderá ser uma imagem e um *link* ou então um texto. No caso do primeiro, é mostrada a imagem e quando o utilizador clicar nesta irá para o endereço do *link*, no segundo o é apenas mostrado um texto.

Foram criados dois manuais de integração ([6][7] em anexo no cd) para ajudar a integrar as *push notifications* e *nag screens* nas aplicações.

No código do lado do cliente primeiramente o estagiário fez o pseudo código e depois com um colega da empresa foi elaborada a prova de conceito, que era uma pequena aplicação para *Iphone* que recebia *push notifications* e mostrava a imagem que recebera do *nag screen*. Mais tarde foi melhorado por um colega, para adaptar às aplicações da empresa.

3. Análise de dados

Inicialmente foi feita uma análise de dados e criado um modelo utilizando a ferramenta de “MySQL Workbench 6.0 CE”. Este modelo foi impresso e foram feitas algumas alterações. Por sua vez este modelo foi passado para o “*SQL Server 2014 Management Studio*” e através deste modelo foi gerado o modelo físico de dados utilizado. Foi utilizada a ferramenta *aspnet_regsql.exe* para que pudesse adicionar o esquema de *SqlMembershipProvider* na base de dados e para que fosse possível utilizar esta tecnologia para autenticação.

Foi criada uma tabela “Empresa” que vai ser uma tabela central à base de dados. Os utilizadores têm de estar associados a esta tabela para realizarem acções. Uma vez que existe uma chave do tipo *uniqueidentifier* que irá identificar todos os utilizadores da empresa, esta chave estará também presente em tabelas únicas para cada empresa para fornecer uma segurança extra no acesso a informação e garantir que apenas utilizadores da empresa têm acesso à informação da sua empresa. Em anexo, em formato digital no cd, a imagem do esquema da base de dados completo.

Em seguida listam-se com uma breve descrição as principais tabelas.

A tabela “*App*” que permite guardar as informações sobre as aplicações como por exemplo o título, SKU, *Apple identifier*, *bundle id* e outros.

Na tabela “*AppDetails*” são guardados os dados que podem variar numa aplicação, por exemplo podem existir várias as linguagens para uma aplicação e nestas línguas variar a descrição, imagem, *link* para a loja e outros.

Podem ainda existir várias versões da mesma aplicação com diferentes dados e para guardar esses dados existe a tabela “*AppVersion*”. Os dados guardados por esta tabela são versão, id da aplicação, data de lançamento e o “*TenantId*”.

A tabela “*Sale*” guarda os dados das vendas feitas, ou melhor dizendo informações dos *downloads* feitos das aplicações para *Iphone* da empresa. Estes dados são obtidos através do *itunes connect* [20]. Os dados guardados são data de início, data de fim uma vez que vários períodos (dia, ano, mês e semana), o número de unidades, lucros do desenvolvedor, preço para o cliente, tipo de produto e outros.

A tabela “*Outsource*” guarda informação sobre as pessoas que trabalharam na aplicação, são guardados o nome, correio electrónico, número de telefone, data em que começou a trabalhar, data em que terminou, id da aplicação em que trabalhou e outros.

A tabela “*NagScreens*” guarda informação sobre os *nagscreens* introduzidos pelo utilizador. Os dados guardados são o “id” da aplicação para o qual o *nagscreen* será enviado, texto, caminho para a imagem a ser mostrada, se está activo, data de início, data de fim e outros.

A tabela “*Screen*” guarda informação dos *more screens* introduzidos pelo utilizador. Os dados guardados são o “id” da aplicação a que o *more screen* pertence, texto, caminho para a imagem a ser mostrada, data de início, data de fim e outros.

A tabela “*InApp*” guarda informação sobre as *inapps*. Uma aplicação poderá ter várias *inapps*. Os dados guardados são o “id” da aplicação à qual a *inapp* pertence, SKU, *Apple identifier* e outros.

4. Desenvolvimento na Kamae

No desenvolvimento do projecto foi utilizada a metodologia *scrum*, onde são feitos *sprints* com duração de duas ou de quatro semanas.

4.1. Aplicação de metodologia *Scrum* no estágio

Durante a realização do estágio a primeira coisa a ser feita foi a realização de um documento de visão/requisitos que posteriormente foi validado.

Na kamae é utilizada a metodologia de *scrum* para o desenvolvimento de *software*. Diariamente, é feita uma reunião onde cada desenvolvedor diz o que fez, o que irá fazer e os desafios encontrados. Esta reunião normalmente é realizada à tarde por volta das 17h30min e nela participa um membro da equipa de do suporte e a equipa de desenvolvimento. Esta reunião traz várias vantagens, na medida em que fornece mais informação sobre o avanço dos projectos à gestão da empresa e vai permitir que as decisões possam ser tomadas mais cedo. Para além disso os colegas que estão noutros projectos também ficam com um pouco de noção do que está a ser feito e muitas vezes são encontradas soluções para desafios uma vez que membros da equipa de desenvolvimento já podem ter passado por situações semelhantes ou podem ter conhecimento que ajude a resolver a situação. Os *sprintes* duram entre 2 ou 4 semanas e para cada um são definidos objectivos. Semanalmente é feito um relatório, onde são indicados os êxitos obtidos (objectivos cumpridos), as dificuldades e objectivos da próxima semana. À segunda-feira é recebido um *email* com os objectivos semanais.

No que diz respeito à qualidade de *software* existe um elemento da equipa que realiza os testes e caso encontre bugs reporta-os no *TFS* para que este seja corrigido. As correcções normalmente são feitas pela pessoa que implementou a aplicação.

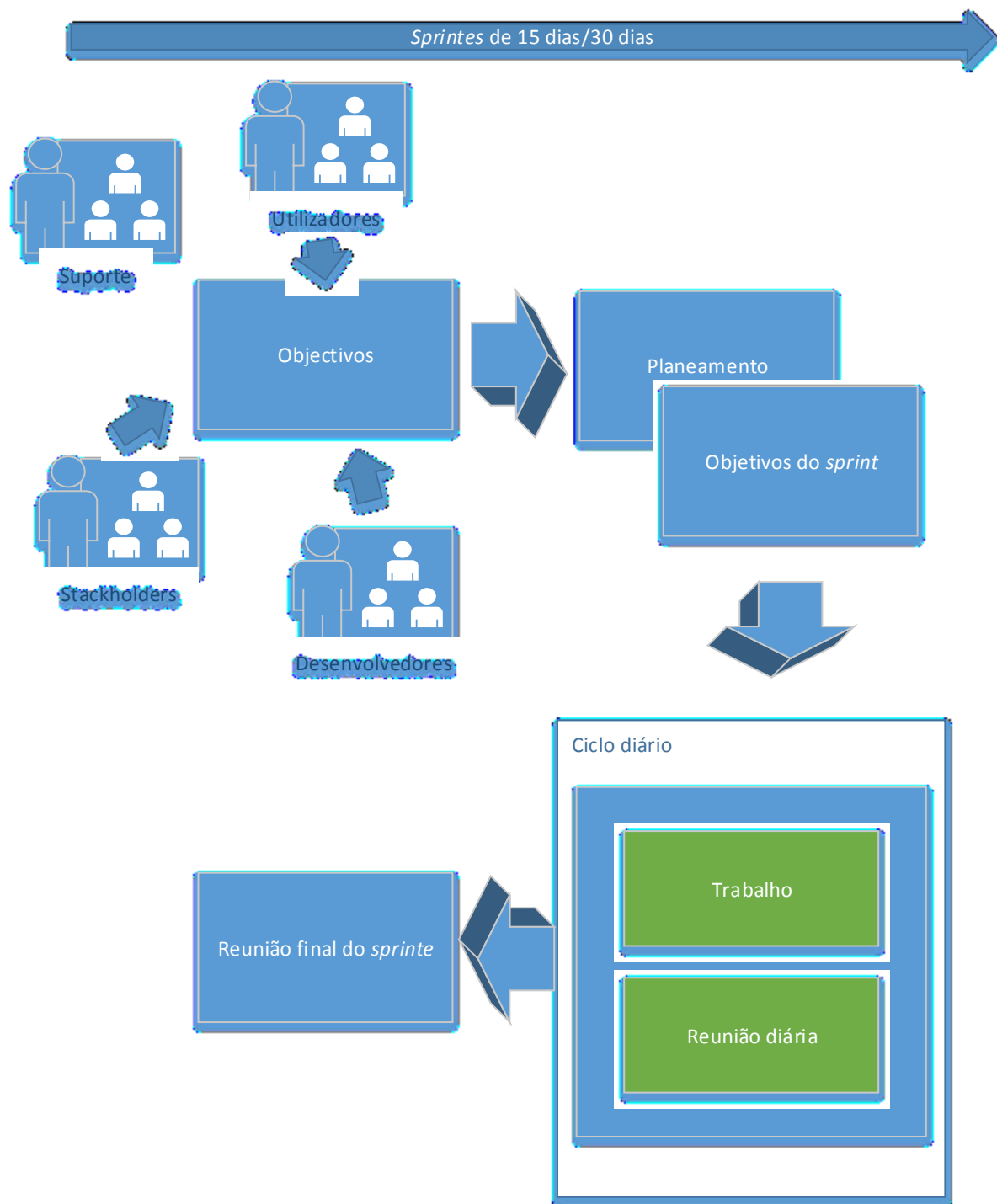


Figura 43 - Esquema de metodologia *scrum* no estágio

4.2. Ferramentas

Na Kamae são utilizadas ferramentas poderosas para o desenvolvimento de *software* podendo ser dado o exemplo o *Cocos* [21] e o *Share Point*. Estas ferramentas permitem fazer *software* mais específico mas com grande velocidade de desenvolvimento.

A principal ferramenta de trabalho é o *Visual Studio*. Um exemplo das vantagens de trabalhar com a mesma ferramenta é o facto de um desenvolvedor poderá ter de pegar num projecto de outro elemento, o que será mais simples se este já estiver habituado a trabalhar com a mesma ferramenta (ambiente de desenvolvimento).

4.3. Partilha de conhecimentos

É feita a partilha de conhecimentos em sessões que existem (*Zen Tech*) em que um membro da Kamae apresenta uma tecnologia, ferramenta ou forma de trabalho aos restantes elementos.

O estagiário participou nas seguintes sessões do *Zen Tech*:

- Cocos 2D.
- Programação *Voip*.
- Sharepoint: O que é? Arquitectura. Vantagens e desvantagens. O que se pode fazer? *Case Studies*.
- Ferramenta de Teste DevExpress: Introdução a testes de *software*.

5. Conclusões

A realização do estágio permitiu ao estagiário adquirir novos conhecimentos não só de cariz tecnológico mas também a nível de metodologia de trabalho e ao nível pessoal o que lhe permitiu conhecer novas pessoas e pontos de vista.

A possibilidade de estagiar numa empresa torna-se uma experiência enriquecedora, principalmente para entender como funciona o desenvolvimento de *software* neste ambiente.

5.1. Análise do desenvolvimento da aplicação IBO

O estágio permitiu experienciar o ciclo de desenvolvimento dentro de uma empresa, o levantamento e especificação de requisitos, a participação em reuniões, entregas do projecto, o desenvolvimento, a correcção de bugs e a cooperação com colegas.

Houveram alguns desafios a superar dos quais a inexperiência com as tecnologias a serem utilizadas, o que demonstrou ser um grande desafio apesar de possuir algum conhecimento nas linguagens *Visual Basic* e *java* que são muito idênticas ao *C#*, mas existem algumas diferenças significativas. As maiores dificuldades foram na *interface* uma vez que no *razor* tem de ser tudo feito com codificação sem o arrasto de componentes, e ainda na codificação de *javascript* e *AJAX* em quais o estagiário não tinha qualquer experiência. Também a integração das diversas tecnologias foi um desafio. Através de algum treino e de desenvolvimento de mostras de conceito foi possível adquirir os conhecimentos que seriam necessários para o desenvolvimento do IBO.

5.2. Metodologia

Num mundo que se tem de reagir rápido e manter o cliente satisfeito, melhorando constantemente o software, é importante manter os projectos sob controlo e não se pode perder tempo. A Kamae parece ter adoptado uma estratégia em que vai não só adoptando melhores técnicas para reagir as suas necessidades, mas também mantendo os seus projectos controlados.

5.3. Trabalho futuro

A aplicação poderá ser incrementada adaptando as funcionalidades que já existem para as aplicações iOS para outras plataformas, por exemplo, o android/Google play, uma vez que também se encontram a ser desenvolvidas aplicações para esta plataforma.

As formas de lucrar com as aplicações *free* são através de publicidade e em vendas de *inapps*. Para isso existem notificações a lembrar o utilizador que a aplicação existe para que este volte à aplicação e veja publicidade e possa comprar *inapps*. As próprias aplicações com *more screens* e *nag screens* facilitam com que o utilizador possa encontrar uma aplicação mais apetecível e a possa adquirir. Ou seja quanto mais vezes o utilizador jogar um jogo mais hipótese de se obter algum lucro. Então pode dizer-se que é objectivo do criador da aplicação tornar esta o mais apetecível possível para que o utilizador a use o máximo possível e para isso, sugeria a realização de um estudo sobre como tornar a aplicação mais apetecível

começando por pesquisar as heurísticas de Malone e Lepper [22] e em seguida criar uma aplicação com os aprendizados do estudo.

A metodologia de desenvolvimento de *software* seguida na Kamae parece ser adequada ao tipo de mercado que serve e em que se tem de reagir rápido a pedidos de clientes e possíveis *bugs* que possam surgir. No entanto, acho que ainda há espaço para melhorias, uma das quais poderia ser a implementação de *pair programming* [23] nalguns dias do mês e em aplicações, ou muito específicas (por exemplo para plataformas android/iphone), ou em aplicações que se encontrem a ser desenvolvidas por uma só pessoa e assim difundir o conhecimento por todos.

Acho que seria importante definir processos de desenvolvimento de *software* pois a sua definição, permitira uma melhor compreensão dos mesmos e uma vez que este possa ser visto penso que será mais fácil alterar e melhorar, embora o *scrum* encoraje as pessoas a pensar/dizer o que acham que se deve melhorar penso que a possibilidade de olhar o que se pode melhorar seria muito mais encorajador e poderiam ser encontradas mais melhorias. Embora se possa passar conhecimento entre as pessoas há sempre pormenores que são esquecidos no passa a palavra, sendo importante na opinião do estagiário que exista um documento que possa transmitir este conhecimento.

Seria interessante a definição de convenções de código para a equipa de desenvolvimento, para que o código pertença a todos para que o código seja mais fácil de ler pelos outros membros da equipa, mais fácil de manter e de reaproveitar.

Seria interessante procurar e escolher uma ferramenta que permitissem desenvolver aplicações multiplataforma de maneira a manter o código todo centralizado e assim os funcionários só tinham de aprender uma ferramenta e quando desenvolvessem para uma plataforma estavam a desenvolver para as outras.

Bibliografia

- [1] <http://www.ecb.europa.eu/stats/eurofxref/eurofxref-daily.xml> [acedido em Dezembro de 2014].
- [2] <http://themoneyconverter.com/rss-feed/EUR/rss.xml> [acedido em Dezembro de 2014].
- [3] “iTunes, App Store, iBooks, and Mac App Store Affiliate Resources - Search API” [online]. <http://www.apple.com/itunes/affiliates/resources/documentation/itunes-store-web-service-search-api.html> [acedido em Dezembro de 2014].
- [4] “Chartboost”, [online]. <https://dashboard.chartboost.com/api-explorer> [acedido em Dezembro de 2014].
- [5] <https://itunes.apple.com/search?term=KAMAESI&entity=software&attribute=softwareDeveloper> [acedido em Dezembro de 2014].
- [6] Manual de integração nag screens.docx
- [7] Manual de integração Push notifications.docx
- [8] “SqlMembershipProvider Class (System.Web.Security)”[online]. [http://msdn.microsoft.com/en-us/library/system.web.security.sqlmembershipprovider\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.web.security.sqlmembershipprovider(v=vs.110).aspx) [acedido em Dezembro de 2014].
- [9] “Creating the Membership Schema in SQL Server (C#) | The ASP.NET Site”, [online] <http://www.asp.net/web-forms/tutorials/security/membership/creating-the-membership-schema-in-sql-server-cs>. [acedido em Dezembro de 2014].
- [10] “Introdução ao ASP.NET Razor | MSDN”, [online]. <http://msdn.microsoft.com/pt-br/library/gg675215.aspx>. [acedido em Dezembro de 2014].
- [11] “Fundamentos do Entity Framework 4”, [online]. <http://msdn.microsoft.com/pt-br/library/jj128157.aspx> [acedido em Dezembro de 2014].
- [12] “JSON”, [online]. <http://json.org/json-pt.html> [acedido em Dezembro de 2014].
- [13] “ScottGu's Blog - ASP.NET MVC 3: Layouts with Razor”, [online]. <http://weblogs.asp.net/scottgu/asp-net-mvc-3-layouts> [acedido em Dezembro de 2014].
- [14] “FormsAuthentication Class (System.Web.Security)”, [online]. [http://msdn.microsoft.com/en-us/library/yh26yfzy\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/yh26yfzy(v=vs.100).aspx) [acedido em Dezembro de 2014].
- [15] “Redth/PushSharp · GitHub”, [online]. <https://github.com/Redth/PushSharp> [acedido em Dezembro de 2014].

- [16] D. Greenwell, "Metro JS - Metro Live Tiles and more Modern UI tools for jQuery - Drew Greenwell - Professional Developer", [online]. <http://www.drewgreenwell.com/projects/metrojs> [acedido em Dezembro de 2014].
- [17] Wenzhixin, "Multiple Select", [online]. <http://wenzhixin.net.cn/p/multiple-select/docs/#examples> [acedido em Dezembro de 2014].
- [18] "Local and Remote Notification Programming Guide: Apple Push Notification Service", [online]. <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html> [acedido em Dezembro de 2014].
- [19] "Developer Controls and Design Tools - .Net Components & Controls - www.Infragistics.com" [online]. <http://www.infragistics.com/> [acedido em Dezembro de 2014].
- [20] "iTunes Connect Sales and Trends Guide: App Store" [versão consultada de 17 de Outubro de 2013]
- [21] "Cocos2d-x: World's #1 Open Source Game Development Platform" [online] <http://www.cocos2d-x.org/> [acedido em Dezembro de 2014].
- [22] "A taxonomy of motivation and game design | Instructional Design Fusions" [online] <https://instructionaldesignfusions.wordpress.com/2011/08/20/a-taxonomy-of-motivation-and-game-design/> [acedido em Dezembro de 2014].
- [23] "Agile Development Methods - Pair Programming - Programmer Productivity - VersionOne" [online] <http://www.versionone.com/agile-101/pair-programming.asp> [acedido em Dezembro de 2014].
- [24] "Entity Framework Querying/Finding Entities" [online] <http://msdn.microsoft.com/en-us/data/jj573936.aspx> [acedido em Dezembro de 2014].
- [25] "ASP.NET MVC Helper Sample - ASP.NET MVC Helper with Grid Control - Ignite UI™" [online] <http://www.igniteui.com/grid/aspnet-mvc-helper> [acedido em Dezembro de 2014]
- [26] "RegExr: Learn, Build, & Test RegEx" [online] <http://www.regexr.com/> [acedido em Dezembro de 2014]
- [27] "Ajax | MDN" [online] <https://developer.mozilla.org/en-US/docs/AJAX> [acedido em Dezembro de 2014]
- [28] "Apple Push Notification Service" [online] <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html> [acedido em Dezembro de 2014]
- [29] "KAMAE - O Melhor amigo do seu negócio! > KAMAE RT" [online] <http://www.kamae.pt/KAMAERT.aspx> [acedido em Dezembro de 2014]

- [30][online] <http://www.isec.pt/isec/> [acedido em Dezembro de 2014]
- [31]" Get Started" [online] <http://www.visualstudio.com/get-started/overview-of-get-started-tasks-vs> [acedido em Dezembro de 2014]
- [32]" Team Foundation Server 2013" [online] <https://msdn.microsoft.com/en-us/vstudio/ff637362.aspx> [acedido em Dezembro de 2014]
- [33]" MVC Architecture" [online] http://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm [acedido em Dezembro de 2014]
- [34]" jQuery" [online] <http://jquery.com/> [acedido em Dezembro de 2014]
- [35]" LINQ (Language-Integrated Query)" [online] <https://msdn.microsoft.com/en-us/library/bb397926.aspx> [acedido em Dezembro de 2014]
- [36]" Lambda Expressions (C# Programming Guide)" [online]<https://msdn.microsoft.com/en-us/library/bb397687.aspx> [acedido em Dezembro de 2014]
- [37]" XML Introduction - What is XML?" [online] http://www.w3schools.com/xml/xml_what_is.asp [acedido em Dezembro de 2014]
- [38]" CSS Introduction" [online] http://www.w3schools.com/css/css_intro.asp [acedido em Dezembro de 2014]
- [39]" Visual C#" [online] <https://msdn.microsoft.com/pt-BR/library/kx37x362.aspx> [acedido em Dezembro de 2014]

